

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



PROYECTO FIN DE CARRERA
INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN:
IMAGEN Y SONIDO

**FRONT-END FOR A BIOMETRIC EXTENDED
EXPERIMENT PLATFORM (BEEP)**

AUTOR: DIEGO CARRERO FIGUEROA
TUTORA: BELÉN RUÍZ MEZCUA
DIRECTOR: LUIS PUENTE RODRÍGUEZ

Julio de 2009

Proyecto Fin de Carrera

Front-End for a Biometric Extended Experiment Platform (BEEP)

AUTOR:

Diego Carrero Figueroa

TUTORA:

Belén Ruíz Mezcua

DIRECTOR:

Luis Puente Rodríguez

La defensa del presente Proyecto Fin de Carrera se realizó el día 17 de Julio de 2009 y fue evaluada por el siguiente tribunal:

PRESIDENTE: Ángel García Crespo

SECRETARIO: Vicente Palacios Madrid

VOCAL: Francisco Javier Calle Gómez

Este trabajo está dedicado a Merche
y a mis padres, Ignacio y M^a Carmen.

Diego.

PRÓLOGO

En los últimos años, la demanda de sistemas fiables de identificación humana ha sufrido un importante incremento debido a la aparición de un gran número de aplicaciones, tanto civiles como militares. Este crecimiento ha estimulado el interés de la comunidad científica en la evaluación de los sistemas biométricos de forma controlada.

Existen tres problemas principales derivados de la evaluación de los sistemas biométricos. En primer lugar, la comparación de resultados de diferentes investigaciones no es rigurosa si el marco de trabajo no es común a la ejecución de todos los experimentos. En segundo lugar, los resultados presentados no son comparables si la información utilizada en las evaluaciones de los sistemas no procede de la misma base de datos. Por último, la realización de experimentos requiere la gestión de un amplio conjunto de tareas para adecuar el entorno de ejecución a las necesidades de los mismos.

Por estos motivos, se hace necesario un contexto de experimentación repetible y bien definido que permita comparar las prestaciones de sistemas biométricos.

A lo largo de la presente memoria se describe el desarrollo del proyecto del Front-End para la plataforma BEEP (Biometric Extended Experiment Platform). Ésta establece un marco controlado de experimentación que permite tanto la creación, edición y ejecución de experimentos como la obtención y comparación de resultados sobre bases de datos biométricos comunes.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	1
1.1. MOTIVACIÓN	1
1.2. OBJETIVOS.....	2
1.3. ESTRUCTURA DEL DOCUMENTO	2
ESTADO DEL ARTE	3
2.1. INTODUCCIÓN	3
2.2. PROCESO BIOMÉTRICO	4
2.2.1. Etapas.....	5
2.2.2. Operaciones.....	6
2.3. BIOMETRÍA UNIMODAL	6
2.3.1. Huella Dactilar.....	7
2.3.2. Geometría Facial	7
2.3.3. Iris	8
2.3.4. Geometría de la Palma de la Mano	8
2.3.5. Huella de la Palma de la Mano	9
2.3.6. Firma.....	9
2.3.7. Voz.....	9
2.3.8. Dinámica del Tecleo	10
2.3.9. Movimiento.....	10
2.4. BIOMETRÍA MULTIMODAL.....	11
2.5. EVALUACIÓN DE LOS SISTEMAS BIOMÉTRICOS.....	13
2.5.1. Iniciativas	13
2.5.2. Tipos de Evaluaciones	14
2.5.3. Medidas de Rendimiento	16
METAMODELO	19
3.1. INTRODUCCIÓN.....	19
3.2. OBJETIVO	20
3.3. CONTEXTO ALGEBRAICO.....	21
3.3.1. Identidad (\mathbb{I}).....	22
3.3.2. Proyección (π).....	22
3.3.3. Selección (σ)	23
3.3.4. Asociación (\mathbb{J})	23
3.3.5. Agrupación (\mathbb{G})	23
3.3.6. Procesado (\mathbb{P})	24
3.4. DEFINICIÓN	24
3.4.1. Interpretación	25
3.4.2. Semántica	26
3.4.3. Sintaxis	26
PLATAFORMA BEEP	31
4.1. INTRODUCCIÓN.....	31
4.2. ESPECIFICACIÓN DE REQUISITOS.....	32
4.2.1. Requisitos de Capacidad	32
4.2.2. Requisitos de Restricción.....	34
4.3. ARQUITECTURA	34
4.4. MODELO DE DATOS.....	36

4.4.1. Modelo de Datos del Sistema	37
4.4.2. Modelo de Datos de la Entidad Investigadora.....	39
4.5. INTERFACE DE DATOS	41
4.6. INTERFACE DE COMUNICACIÓN	43
4.7. DECISIONES TECNOLÓGICAS	47
FRONT-END. ANÁLISIS.....	51
5.1. INTRODUCCIÓN	51
5.2. IDENTIFICACIÓN DE STAKEHOLDERS	51
5.3. DEFINICIÓN DEL SISTEMA	52
5.4. USUARIOS DEL FRONT-END	52
5.4.1. Perfiles de Usuario.....	52
5.4.2. Roles de Usuario	53
5.4.3. Asociación de Perfiles y Roles	53
5.5. CASOS DE USO	54
5.5.1. Diagramas de Casos de Uso	54
5.5.2. Descripción de Casos de Uso.....	59
FRONT-END. DISEÑO	85
6.1. INTRODUCCIÓN	85
6.2. ARQUITECTURA	85
6.2.1. Arquitectura Física	85
6.2.2. Arquitectura Lógica	85
6.3. CAPA DE NEGOCIO	87
6.4. CAPA DE PRESENTACIÓN.....	89
6.4.1. Mapa Conceptual.....	89
6.4.2. Diseño Gráfico	91
6.5. EDITOR DE EXPERIMENTOS.....	94
GESTIÓN DEL PROYECTO	97
7.1. INTRODUCCIÓN	97
7.2. METODOLOGÍA	98
7.3. ESTIMACIÓN DE RECURSOS TEMPORALES	99
7.4. ESTIMACIÓN DE RECURSOS ECONÓMICOS	101
7.4.1. Recursos Materiales	101
7.4.2. Recursos Humanos	101
7.4.3. Costes Totales	103
7.5. PLAN DEL PROYECTO	104
7.5.1. Fases del Proyecto	104
7.5.2. Control de Modificaciones.....	106
7.5.3. Seguimiento del Proyecto	106
7.6. HERRAMIENTAS	106
CONCLUSIONES Y TRABAJO FUTURO	109
8.1. CONCLUSIONES	109
8.2. TRABAJOS FUTUROS.....	110
REFERENCIAS.....	111
ANEXO A. GLOSARIO	119
ANEXO B. ESPECIFICACIÓN DE REQUISITOS	121
ANEXO C. SEGUIMIENTO DEL PROYECTO	129

ÍNDICE DE FIGURAS

Figura 2.1. Diagrama de Flujo del Proceso Biométrico	5
Figura 2.2. Extracción de Características de la Huella Dactilar	7
Figura 2.3. Extracción de Características de la Geometría Facial.....	8
Figura 2.4. Extracción de Características de Iris	8
Figura 2.5. Extracción de Características de la Geometría de la Mano	9
Figura 2.6. Extracción de Características de la Palma de la Mano	9
Figura 2.7. Extracción de Características de la Voz.....	10
Figura 2.8. Extracción de Características de la Silueta	11
Figura 2.9. Escenarios Operacionales de los Sistemas Multimodales	12
Figura 2.10. Curvas de Falso Rechazo y Falsa Aceptación	16
Figura 2.11. Curva ROC.....	17
Figura 3.1. Marco de Evaluación de un Sistema Biométrico	21
Figura 3.2. Definición Algebraica de un Experimento.....	22
Figura 3.3. Primitiva Identidad.....	22
Figura 3.4. Primitiva Proyección.....	23
Figura 3.5. Primitiva Selección.....	23
Figura 3.6. Primitiva Asociación.....	23
Figura 3.7. Primitiva Agrupación.....	24
Figura 3.8. Primitiva Proceso.....	24
Figura 3.9. Ejemplo del Modelo de un Experimento.....	25
Figura 3.10. Diagrama de Clases del Metamodelo	26
Figura 3.11. Elementos Visuales Asociados a las Operaciones.....	27
Figura 3.12. Definición Algebraica del Nodo Tap	28
Figura 3.13. Definición Algebraica del Nodo Join.....	28
Figura 3.14. Definición Algebraica del Nodo Group	28
Figura 3.15. Definición Algebraica del Nodo Process	29
Figura 4.1. Visión General de la Plataforma.....	32
Figura 4.2. Arquitectura Física de la Plataforma	35
Figura 4.3. Arquitectura Lógica de la Plataforma	36
Figura 4.4. Diagrama E-R para el Modelo de Datos del Sistema.....	37
Figura 4.5. Diagrama E-R para el Modelo de Datos de la Entidad.....	39
Figura 4.6. Estado Inicial del Interface de Comunicación.....	45
Figura 4.7. Estado Intermedio del Interface de Comunicación	46
Figura 4.8. Estado de Indisposición del Interface de Comunicación	47
Figura 5.1. Diagrama de Casos de Uso para la Gestión de Usuarios	56
Figura 5.2. Diagrama de Casos de Uso para la Gestión de Entidades	56
Figura 5.3. Diagrama de Casos de Uso para la Gestión de Experimentos	57
Figura 5.4. Diagrama de Casos de Uso para la Gestión de Procesos	57
Figura 5.5. Diagrama de Casos de Uso para la Gestión de Almacenes	58
Figura 6.1. Diagrama de Componentes del Front-End.....	86
Figura 6.2. Paquetes de la Capa de Negocio.....	88
Figura 6.3. Nivel Superior del Interface Gráfico.....	90
Figura 6.4. Segundo Nivel del Interface Gráfico.....	90
Figura 6.5. Nivel Inferior del Interface Gráfico	91
Figura 6.6. Diseño Gráfico de las Páginas Públicas.....	93
Figura 6.7. Diseño Gráfico de las Páginas Privadas	93
Figura 6.8. Paquetes del Editor de Experimentos	95

Figura 7.1. Proceso de Desarrollo Software	97
Figura 7.2. Ciclo de Vida en Espiral	98
Figura 7.3. Diagrama de Gantt	100
Figura 7.4. Expresión para el Cálculo de las Amortizaciones	101

ÍNDICE DE TABLAS

Tabla 2.1. Características Personales como Identificadores Biométricos	4
Tabla 4.1. Requisitos de Capacidad de la Plataforma.....	33
Tabla 4.2. Requisitos de Restricción de la Plataforma	34
Tabla 5.1. Contexto Formal para los Perfiles y Roles de Usuario	54
Tabla 7.1. Recursos Temporales por Fases del Proyecto.....	99
Tabla 7.2. Recursos Materiales para el Desarrollo del Proyecto.....	101
Tabla 7.3. Recursos Humanos	102
Tabla 7.4. Costes del Proyecto	103
Tabla 7.5. Recursos Materiales Proporcionados por el Cliente.....	103

INTRODUCCIÓN

Este primer capítulo recoge las motivaciones que han llevado a la creación de la plataforma y ofrece los objetivos buscados con el desarrollo de la misma.

1.1. MOTIVACIÓN

Dentro del grupo SoftLab, perteneciente al Departamento de Informática de la Universidad Carlos III de Madrid, se planteó el desarrollo de una plataforma de experimentación biométrica que permitiese la investigación y la obtención de resultados de manera fiable bajo un mismo entorno. Las líneas de investigación del grupo contemplan la evaluación y el desarrollo de sistemas de reconocimiento biométrico de personas.

Las tecnologías que soportan los sistemas biométricos se han convertido en uno de los principales puntos de atención por parte de muchas empresas y entidades. Como consecuencia, la investigación ha crecido notablemente en los últimos años en un intento de ofrecer implementaciones de sistemas más precisos y seguros.

Para conseguir un entorno de pruebas controlado, diferentes organizaciones comerciales y la comunidad científica han desarrollado competiciones para evaluar sistemas biométricos, pero ninguna de ellas ha propuesto el desarrollo de una plataforma como la planteada en este proyecto.

En la evaluación de los sistemas biométricos se emplean normalmente bases de datos recolectadas en medios físicos. Su disponibilidad supone una limitación a la hora de evaluar las prestaciones de un sistema y la comparación de resultados no es rigurosa si no se dispone de la misma información biométrica.

El tratamiento de la información contenida en las bases de datos se debe ajustar al cumplimiento de la Ley Orgánica de Protección de Datos de Carácter Personal [LOPD, 1999]. Actualmente, la distribución y el propio formato de las bases de datos biométricos impide a los usuarios donantes ejercitar el derecho a la retirada de su información personal.

1.2. OBJETIVOS

El objetivo de la plataforma es solucionar los problemas derivados de la experimentación biométrica. A partir de las motivaciones expuestas, se propone un marco de trabajo para los investigadores que resulte sencillo, repetible, modificable y donde la información biométrica se encuentre protegida del libre acceso. Esto se traduce en la implementación de un entorno estable e inalterable de experimentación que permita la obtención de resultados fiables y lo más importante de todo, su comparación.

El objetivo del proyecto, derivado de los expuestos en el párrafo anterior, es el desarrollo del Front-End de la plataforma y del metamodelo para describir el proceso de ejecución de experimentos.

1.3. ESTRUCTURA DEL DOCUMENTO

La presente memoria recoge la evolución del proyecto en las próximas secciones. En lo siguiente, este documento está formado por siete capítulos. El capítulo 2 recoge el Estado del Arte de los sistemas biométricos y presenta los diferentes marcos de experimentación. El capítulo 3 expone el metamodelo que da soporte conceptual a la plataforma y que establece la modelización de los experimentos biométricos. El documento continúa con el capítulo 4, el cuál presenta los requisitos y las limitaciones impuestas a la plataforma. Los capítulos 5 y 6 recogen el análisis y el diseño respectivamente para el Front-End. A continuación, el capítulo 7 muestra la gestión del proyecto y finalmente, el capítulo 8 ofrece las conclusiones y las líneas futuras de trabajo.

ESTADO DEL ARTE

A lo largo de este capítulo se presenta una visión general de la biometría. En primer lugar, se describen las etapas del proceso biométrico, a continuación se enumeran las principales tecnologías y finalmente, se muestran los diferentes esquemas de experimentación y las medidas de evaluación más utilizadas.

2.1. INTRODUCCIÓN

La biometría es el estudio mensurativo o estadístico de los fenómenos o procesos fisiológicos. Es un término que acompaña a la aplicación de métodos estadísticos en la medición de procesos biológicos [Dessimoz et al, 2005]. La biometría hace referencia a todas aquellas tecnologías que se emplean para analizar las características fisiológicas y de comportamiento de los individuos.

Se consideran rasgos útiles para los objetivos de la biometría aquellos que son susceptibles de ser medidos o caracterizados mediante parámetros computables. En la Tabla 2.1 se muestran los identificadores más utilizados en la actualidad. De manera ideal, la parametrización de cualquier rasgo debería satisfacer las siguientes propiedades [Jain et al, 2004]:

- Disponibilidad: El rasgo debería estar presente en todos los sujetos de la población.
- Diferenciación: Su variación debería ser elevada entre los diferentes sujetos de la población.
- Invariabilidad: Su variación debería ser mínima para un único sujeto de la población.
- Accesibilidad: La captura de la información del rasgo debería ser una operación sencilla.

Sin embargo, en un sistema biométrico existen otras cuestiones que deben ser consideradas [Jain et al, 2004]:

- Rendimiento: Hace referencia tanto a la precisión y velocidad del proceso de reconocimiento como a los recursos necesarios para conseguir los niveles de rendimiento deseados.

- Aceptabilidad: Expresa la capacidad de aceptación de un determinado rasgo biométrico por parte de la usuarios.
- Elusión: Refleja la facilidad con la que el sistema puede ser engañado mediante métodos fraudulentos.

TIPO	RASGOS BIOMÉTRICOS
Fisiológicos	<ul style="list-style-type: none"> • Voz. • Huella dactilar. • Geometría facial. • Análisis del iris. • Geometría de la palma de la mano. • Huella de la palma de la mano. • Venas del dorso de la mano.
De comportamiento	<ul style="list-style-type: none"> • Voz¹. • Firma. • Dinámica de tecleo. • Movimiento y/o cadencia del paso.

Tabla 2.1. Características Personales como Identificadores Biométricos

El concepto de reconocimiento automático de personas pertenece a los últimos años de la década de los setenta [Raphael et al, 1974]. Los rasgos biométricos presentados en la Tabla 2.1 son características intrínsecas de las personas y por lo tanto, pueden ser empleados en aplicaciones relacionadas con el reconocimiento de la identidad. En los últimos años se han desarrollado ampliamente las tecnologías de los sistemas biométricos y el número de implementaciones ha aumentado notablemente. La biometría se encuentra en continua evolución y su futuro promete una industria de costes asequibles y un amplio abanico de aplicaciones.

2.2. PROCESO BIOMÉTRICO

Un sistema biométrico es un proceso que consta de tres fases [Blackburn, 2004]. En un primer paso, uno o varios sensores adaptadores reciben una señal biométrica. A continuación, esta señal se caracteriza mediante un algoritmo o función matemática para obtener un conjunto de datos que se pueden manipular y comparar. En el tercer y último paso, se comparan los datos generados en el paso anterior con la información existentes en una base de datos y se obtiene un resultado que indica si el individuo generador de la

¹ La voz no es sólo un rasgo de comportamiento sino que también posee un componente fisiológico.

señal es aceptado o rechazado.

2.2.1. Etapas

El proceso biométrico consiste en la adquisición y transformación de datos para realizar cualquier tipo de tarea relacionada con la determinación de la identidad [Dessimoz et al, 2005].

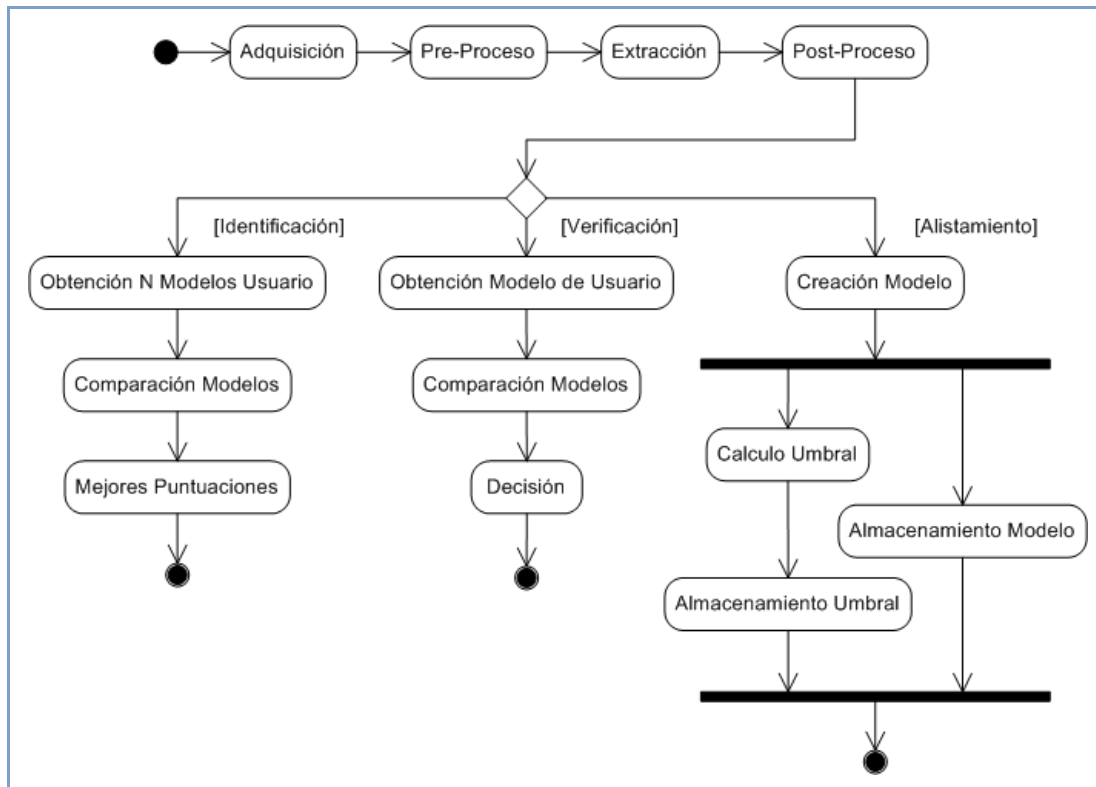


Figura 2.1. Diagrama de Flujo del Proceso Biométrico

En [Dessimoz et al, 2005] se describe el proceso biométrico como una sucesión lineal de operaciones. El diagrama de flujo de la Figura 2.1 [Dessimoz et al, 2005] ofrece una visión esquemática del proceso biométrico. De manera general y para todas las modalidades biométricas, la información se transforma de acuerdo a los siguientes pasos:

- **Captura:** La información se recoge por medio de un sensor y se almacena en memoria.
- **Pre-procesado:** La información adquirida en el dominio de la señal se prepara para la extracción de características. Normalmente, se realiza una normalización como paso previo a la extracción.
- **Extracción de características:** Se obtiene la representación de la información biométrica en un nuevo dominio. Este proceso reduce la dimensionalidad de la señal y crea una representación a nivel de características del patrón de entrada. Esta nueva señal se emplea en el posterior reconocimiento de patrones.

- **Post-procesado:** Las características obtenidas de la señal de entrada se normalizan a fin de adaptarlas al sistema de clasificación.
- **Creación de modelos:** Mediante un conjunto de datos de entrenamiento se genera un modelo matemático para obtener una representación genérica de un determinado usuario. En algunos sistemas es necesario crear un modelo de usuarios de fondo con el fin de normalizar las puntuaciones.
- **Almacenamiento de modelos:** Una vez se han estimado los parámetros que definen la identidad de un usuario, se almacena el modelo de tal forma que pueda ser empleado en etapas posteriores del proceso.
- **Comparación de modelos:** Se compara un conjunto de vectores de características con el modelo biométrico de un determinado usuario. Esta comparación, con uno o varios modelos, resulta en una puntuación que recoge el parecido entre los datos biométricos de entrada y las identidades de los modelos con los que han sido cotejados.
- **Cálculo de umbrales:** A partir de datos de usuarios reales e impostores se calcula un umbral de decisión que marca la aceptación de la identidad en el sistema.

2.2.2. Operaciones

Las etapas descritas anteriormente se emplean en tres operaciones de alto nivel [Dessimoz et al, 2005]: alistamiento, verificación e identificación.

- **Alistamiento:** Con esta operación se añade un nuevo usuario al sistema. Mediante un cierto número de representaciones biométricas se entrena un modelo para el usuario y se adapta el modelo de los usuarios de fondo en caso de ser necesario.
- **Verificación:** La verificación consiste en comparar la representación biométrica de un usuario donante frente al modelo de un usuario pretendido. Una vez contrastada la información con el modelo se obtiene una puntuación que se coteja con un umbral para aceptar o denegar la identidad pretendida.
- **Identificación:** Mediante la identificación se busca la representación biométrica más parecida a la representación de entrada en una base de datos de modelos de usuarios. La información de entrada se compara con todos los modelos de interés y aquel que obtiene la mayor puntuación se sugiere como identidad.

2.3. BIOMETRÍA UNIMODAL

Una vez introducido el concepto de biometría, a continuación se presentan las

características más relevantes de los principales rasgos.

2.3.1. Huella Dactilar

Los sistemas basados en huella dactilar son una de las tecnologías con mayor grado de madurez y actualmente, son los sistemas más utilizados y aceptados a nivel mundial. Estos sistemas tienen su base en los desarrollos realizados por Galton y Purkinje a finales del siglo XVIII. De forma general, la extracción de características de una huella dactilar sigue el esquema presentado en la Figura 2.2.

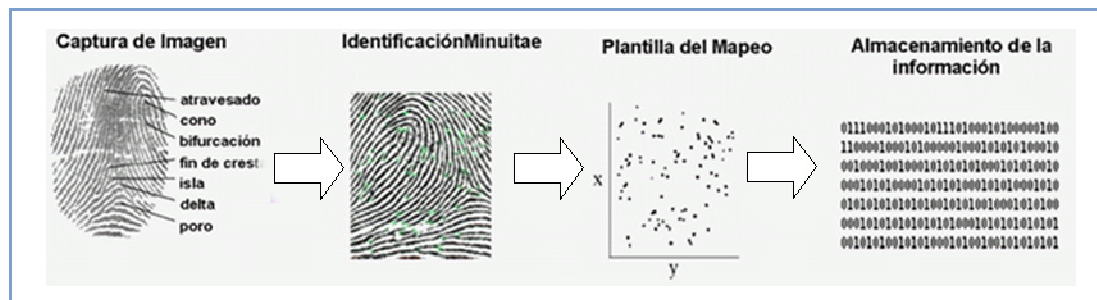


Figura 2.2. Extracción de Características de la Huella Dactilar

La extracción de características en estos sistemas se basa en el análisis de una imagen en escala de grises. Principalmente existen dos procedimientos. Por un lado, en los sistemas basados en correlación [Maltoni et al, 2003] se superponen dos imágenes de la huella dactilar y se calcula la correlación entre ambas para diferentes posiciones mediante traslación y rotación. Por otro lado, en los sistemas basados en la extracción de puntos característicos [Yager et al, 2004] se segmenta la imagen y se localizan puntos singulares (bifurcaciones, fines de cresta, islas,...) denominados minucias, cuya disposición espacial ofrece una elevada capacidad de diferenciación.

2.3.2. Geometría Facial

La mayoría de los sistemas actuales de reconocimiento facial determinan la apariencia de un sujeto a través de la obtención de puntos nodales de la cara. A partir de estos puntos, se determina la distancia entre los ojos, la anchura de la nariz, la distancia del ojo a la boca o la longitud de la línea de la mandíbula [Zhao et al, 2003]. La Figura 2.3 recoge el proceso de extracción de características de la geometría facial.

Aunque existen diversos métodos aplicables al reconocimiento facial, en la actualidad se utilizan principalmente los métodos basados en las características geométricas de la cara.

Land Marks se basa en la localización de diferentes características geométricas como la distancia entre ojos o la anchura de la nariz. En las técnicas de correlación se toman dos imágenes y se calcula el índice de correlación entre ambas para obtener un indicador del parecido. Finalmente, la técnica de Eigen Faces codifica una imagen facial de tal forma que puede ser representada

como la combinación lineal de otras imágenes [Sirovich et al, 1987]. Mientras que las técnicas de correlación imponen un elevado coste computacional, Eigen Faces permite optimizar el sistema y aumenta la velocidad de procesado.

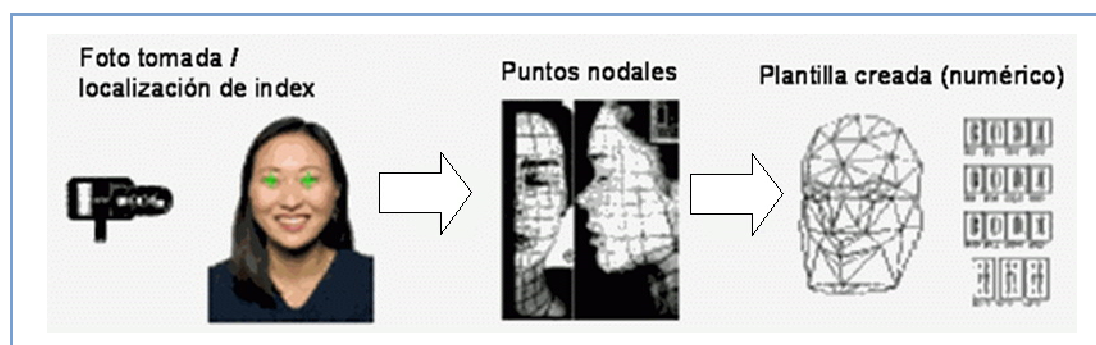


Figura 2.3. Extracción de Características de la Geometría Facial

2.3.3. Iris

El iris es la región anular del ojo que se encuentra limitada por la pupila en la parte interior y por la esclera en la exterior. Sus patrones se forman durante el desarrollo del feto y se estabilizan durante los dos primeros años de vida. En la actualidad se emplean varias aproximaciones al problema del reconocimiento basado en iris [Wildes, 2005].

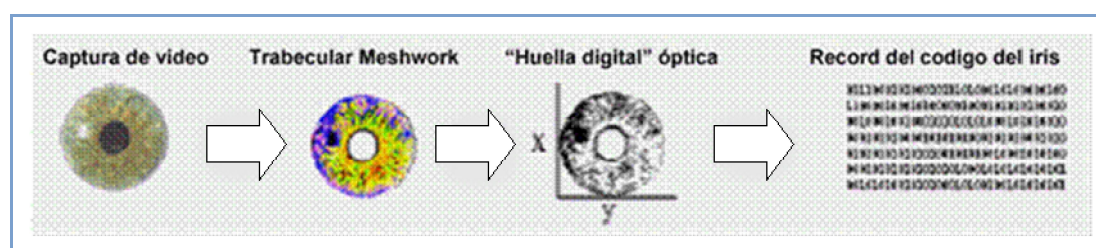


Figura 2.4. Extracción de Características de Iris

Los métodos de ajuste basados en modelos de histogramas procesan la imagen del iris mediante filtros paso-banda de dos dimensiones y utilizan el discriminante lineal de Fisher para realizar el proceso de comparación. Otros sistemas [Tisee, 2003] emplean la transformada de Hough para localizar el iris y mediante la transformada de Hilbert obtienen las características de éste. El último tipo de sistemas [Masek, 2003] utilizan transformadas de Hough circulares para localizar el iris y filtros de Gabor para extraer el conjunto de características.

2.3.4. Geometría de la Palma de la Mano

El reconocimiento biométrico por medio de la geometría de la mano se fundamenta en el cálculo de las distancias entre diferentes puntos característicos de la palma de la mano [Sánchez et al, 2000 y Sánchez, 2000] a partir de una imagen bidimensional de la misma. Entre las medidas obtenidas se incluye normalmente la longitud de los dedos y el ancho y alto de la mano.

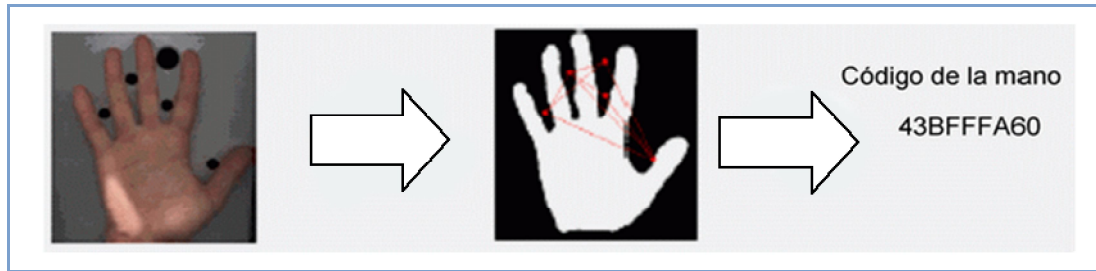


Figura 2.5. Extracción de Características de la Geometría de la Mano

Estos sistemas no están orientados a aplicaciones de seguridad ya que la información obtenida de la geometría de la mano no permite identificar unívocamente a una persona.

2.3.5. Huella de la Palma de la Mano

De la misma forma que la huella dactilar, el reconocimiento mediante la palma de la mano se basa en la información representada por las crestas y valles de su superficie [Duta et al, 2002]. La información de la palma incluye entre otras características el sentido de las crestas y la presencia o ausencia de minucias.

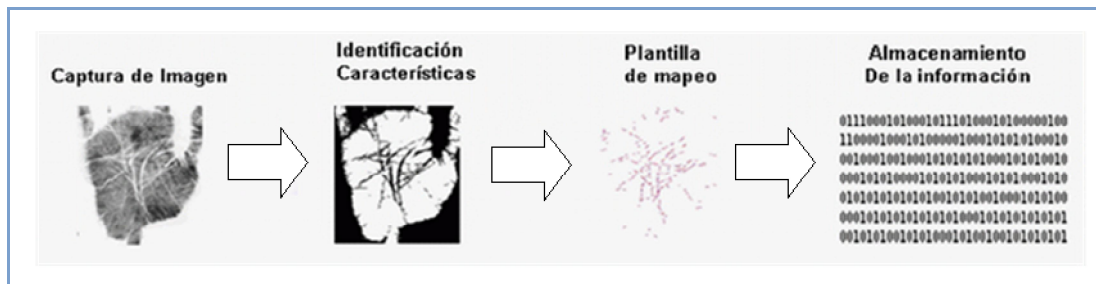


Figura 2.6. Extracción de Características de la Palma de la Mano

2.3.6. Firma

La firma es una característica de comportamiento que se origina por los movimientos rápidos de la mano debidos a la actividad del sistema neuromuscular. El reconocimiento de firma ofrece dos tipos de aproximaciones.

Los métodos on-line obtienen características dinámicas del rasgo biométrico. Los principales métodos de verificación de firma on-line se basan en Dynamic Time Warping [Sato et al, 1982], HMM con distribuciones de probabilidad Gaussianas [Richiardi et al, 2003] y redes neuronales [Chang et al, 1993]. Por su parte, los métodos off-line procesan una imagen para obtener características estáticas de la misma. Algunos de estos sistemas extraen características geométricas de la imagen y emplean redes neuronales para realizar la verificación [Huang et al, 1997]. Otros sistemas [Justino et al, 2001] hacen uso de HMM para caracterizar la distribución espacial de la firma.

2.3.7. Voz

El reconocimiento de personas mediante voz se soporta tanto en la estructura

física del sujeto como en las características de comportamiento del mismo. El esquema de extracción de características se muestra en la Figura 2.7. Existen dos tipos de sistemas [Reynolds, 1995] de reconocimiento. Mientras que los sistemas dependientes del texto se basan en la repetición de un texto específico, los sistemas no dependientes no imponen ningún tipo de restricción en la locución del usuario.

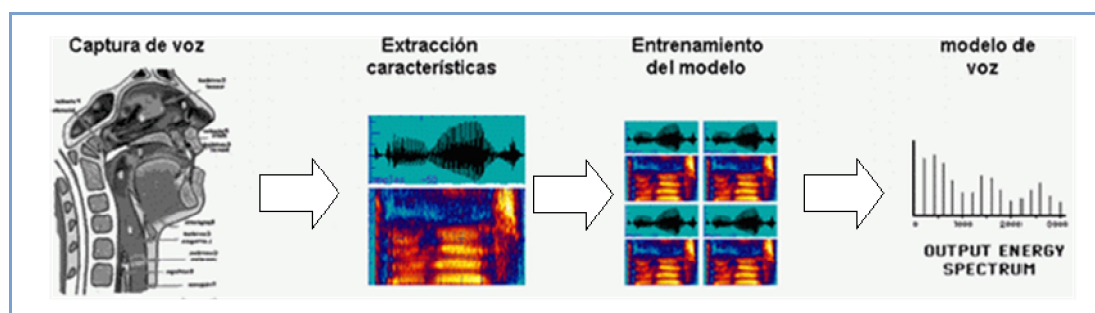


Figura 2.7. Extracción de Características de la Voz

A lo largo de los años se han propuesto diferentes algoritmos para solventar el problema del reconocimiento de locutores [Dessimoz et al, 2005]. Los métodos probabilísticos modelan de forma estadística el comportamiento de la voz [Ruíz, 1998]. A este tipo pertenecen los Modelos Ocultos de Markov [Poritz, 1982] y los Modelos de Mezclas Gaussianas [Reynolds, 1995]. Otros métodos de clasificación empleados en el reconocimiento de locutores son las Redes Neuronales [Oglesby et al, 1988], las Máquinas de Vectores Soporte [Schmidt et al, 1996] y Dynamic Time Warping [Pandit et al, 1998].

2.3.8. Dinámica del Tecleo

Es un tipo de rasgo biométrico conductual empleado en la verificación de la identidad de un individuo mediante su cadencia de escritura en un teclado [Monrose et al, 2000]. Esta tecnología se sostiene sobre la premisa de que cada individuo exhibe un patrón y una cadencia distintivos.

En la mayoría de los sistemas se emplea la latencia entre pulsaciones como característica; sin embargo, en otros se utiliza también el tiempo que permanecen la teclas presionadas. Esta tecnología no requiere hardware o dispositivos adicionales ya que se soporta sobre software de captura. Los principales sistemas se fundamentan en clasificadores bayesianos, redes neuronales y lógica difusa.

2.3.9. Movimiento

El movimiento es un tipo de rasgo biométrico conductual que se usa para verificar la identidad de un individuo examinando su patrón de marcha al caminar [Nixon et al, 2006 y Boyd et al, 2005]. Mediante este rasgo se puede realizar el reconocimiento a larga distancia e incluso con imágenes de baja resolución. Los sistemas se pueden clasificar en estáticos (basados en la figura humana) y dinámicos (basados en el movimiento del individuo).



Figura 2.8. Extracción de Características de la Silueta

2.4. BIOMETRÍA MULTIMODAL

Los sistemas que trabajan utilizando un único rasgo biométrico tienen una serie de limitaciones:

- **Ruido:** La información capturada puede ser ruidosa o puede contener distorsión. Estos hechos son el resultado de la utilización de sensores defectuosos y de condiciones ambientales adversas.
- **Variación intra-clase:** Los datos capturados de un individuo en la operación de autenticación pueden ser muy diferentes de los empleados en la generación de su modelo durante la operación de alistamiento.
- **Variación inter-clase:** La escasa variación de un rasgo biométrico entre los sujetos de una población limita su capacidad de discriminación.
- **Universalidad:** Es posible que un determinado subconjunto de usuarios no posea un rasgo biométrico particular.
- **Ataques de impostores:** Un usuario ilegítimo puede falsificar el rasgo biométrico de otro legítimo para eludir el sistema.

Las limitaciones impuestas por los sistemas unimodales pueden ser solventadas mediante la utilización de múltiples modalidades [Puente et al, 2008]. Los sistemas biométricos multimodales utilizan varios rasgos para realizar el reconocimiento de individuos [Hong et al, 1999].

La Figura 2.9 [Jain et al, 2004] recoge los diferentes escenarios de operación de los sistemas multimodales. Estos sistemas se diseñan para trabajar de cinco formas diferentes y aunque algunas de ellas no implican múltiples modalidades, implican la fusión de información en algunos puntos. Los posibles esquemas de combinación son los siguientes:

- **Un rasgo, múltiples sensores:** Se adquiere un único rasgo mediante diferentes sensores y se combina para mejorar el proceso de reconocimiento.
- **Múltiples rasgos:** Se obtienen y combinan diferentes características

biométricas de una persona.

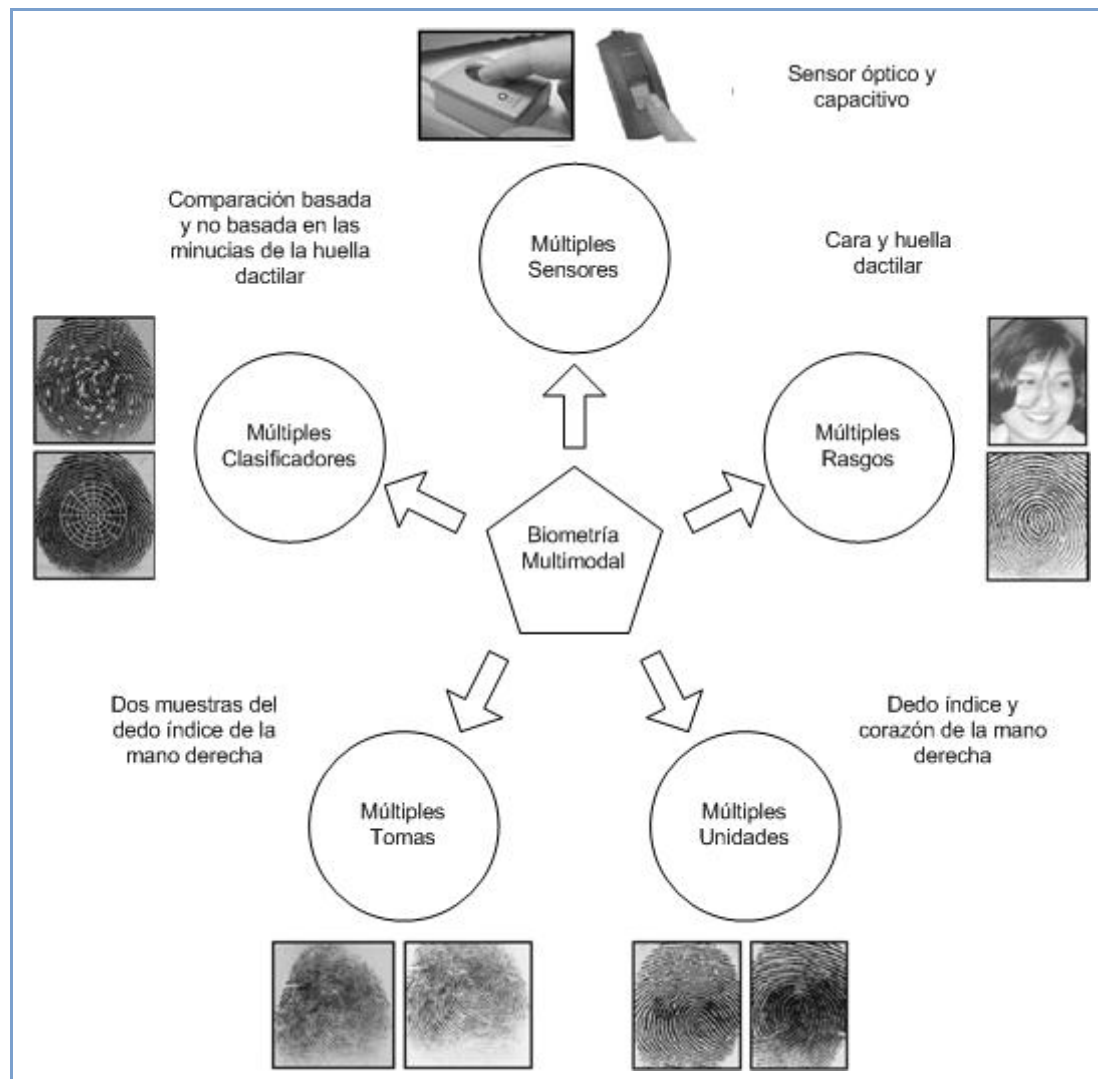


Figura 2.9. Escenarios Operacionales de los Sistemas Multimodales

- Un rasgo, múltiples unidades: Se adquieren y combinan diferentes unidades del mismo rasgo.
- Un rasgo, múltiples representaciones: Se adquieren múltiples tomas de un mismo rasgo por el mismo sensor.
- Un rasgo, múltiples clasificadores: Se obtiene un rasgo a través de un único sensor y mediante diferentes técnicas de extracción de características y/o de comparación de patrones se realiza el proceso de reconocimiento.

A partir de los escenarios descritos y de los componentes de un sistema biométrico existen diferentes posibilidades de fusión [Jain et al, 2004]:

- Fusión a nivel de extracción de características: En este nivel se combinan las características extraídas de los rasgos biométricos en un único vector.

- **Fusión a nivel de puntuaciones:** En este nivel se combinan las puntuaciones que describen las similitudes entre los datos adquiridos y los modelos de cada sistema. Este modo de fusión requiere que las puntuaciones de los subsistemas se encuentren normalizadas en un dominio común [Jain et al, 2005]. La fusión a este nivel es la preferida debido a la facilidad de acceso y de combinación que ofrecen las puntuaciones.
- **Fusión a nivel de decisiones:** En este nivel se combinan las decisiones tomadas por cada sistema con el fin de obtener la decisión final.

2.5. EVALUACIÓN DE LOS SISTEMAS BIOMÉTRICOS

Desde hace varios años, la comunidad científica y las organizaciones comerciales han venido ofreciendo medidas de prestaciones para los sistemas biométricos basadas en bases de datos propias y protocolos de evaluación no estandarizados [Capelli et al, 2006]. Por este motivo, la mayoría de los resultados no son susceptibles de ser comparados y a menudo carecen de importancia.

Un protocolo de evaluación [Phillips et al, 2000] determina tanto la forma en la que se testea un sistema y se miden sus prestaciones como la información utilizada para evaluarlo. En la actualidad, los procesos de evaluación más importantes se encuentran administrados por grupos independientes y la información utilizada para testear los sistemas no ha sido utilizada nunca antes. Para que un protocolo de evaluación sea aceptado por la comunidad científica, los detalles de sus procedimientos han de estar publicados junto con las medidas de prestaciones propuestas y algunos ejemplos representativos del conjunto de datos.

2.5.1. Iniciativas

En los últimos años se han desarrollado diferentes iniciativas dentro de la comunidad científica en un claro esfuerzo por ofrecer un marco común de evaluación para los sistemas biométricos. Las principales iniciativas son las siguientes:

- El Instituto Nacional de Estándares y Tecnología² de Estados Unidos realiza desde hace años diferentes competiciones para evaluar algoritmos y tecnologías de reconocimiento facial [Phillips et al, 2003] y de locutor [Martin et al, 2004].
- La Universidad de Bolonia desarrolla desde el año 2000 una competición bianual, la Fingerprint Verification Competition³, para evaluar el Estado del Arte de los sistemas de verificación basados en huella dactilar [Capelli et

² <http://www.nist.gov/index.html>

³ <http://bias.csr.unibo.it/fvc2006/>

al, 2006].

- El Instituto de Ciencias de la Academia China de las Ciencias realiza desde el año 2004 competiciones para evaluar las prestaciones de los sistemas de verificación de identidad basados en iris. Así mismo, desarrolla la Signature Verification Competition para valorar los sistemas de verificación de identidad basados en firma on-line.
- El Face Recognition Vendor Test⁴ desarrolla competiciones para la evaluación de sistemas comerciales basados en geometría facial desde el año 2000.

2.5.2. Tipos de Evaluaciones

La evaluación de un sistema biométrico requiere la captura de información. Ésta se utiliza tanto en la generación de modelos en la etapa de entrenamiento como en la obtención de puntuaciones en la etapa de reconocimiento [Capelli et al, 2006]. La información capturada puede ser empleada inmediatamente o puede ser almacenada para ser utilizada posteriormente. Los tests de evaluación de los sistemas biométricos se pueden clasificar en dos grupos [Mansfield et al, 2002]:

- Evaluación on-line: La realización de las pruebas requiere la presencia del usuario y su alistamiento o el cálculo de su puntuación se realiza inmediatamente tras la captura de la información biométrica. Como ventaja, el sistema solamente almacena la información estrictamente necesaria puesto que la muestra se elimina tras su utilización. Desafortunadamente, una misma operación no puede ser repetida debido a la falta de la muestra descartada.
- Evaluación off-line: En este segundo tipo, las pruebas se basan en información capturada y almacenada previamente. Este tipo de evaluación requiere un mayor almacenamiento de información pero a cambio permite reproducir un mismo test bajo idénticas condiciones.

Por su parte, los tests off-line se pueden clasificar en uno de los siguientes grupos [Capelli et al, 2006]:

- Test propio en entorno local (In-house – Self-defined): La recolección de los datos biométricos y la definición del protocolo de prueba son realizadas directamente por el investigador. Este entorno no permite realizar ningún tipo de comparación puesto que el test no es reproducible por terceras partes.
- Test predefinido en entorno local (In-house – Existing benchmark): El test se realiza sobre una base de datos de carácter público de acuerdo a protocolos existentes. En este caso, los resultados obtenidos son

⁴ <http://www.frvt.org/>

comparables con otros obtenidos mediante el mismo protocolo y la misma base de datos. El principal inconveniente de este tipo de tests reside en la imposibilidad de predecir las verdaderas prestaciones del sistema en aplicaciones reales debido al posible sobreajuste sobre los datos utilizados.

- Test independiente y débilmente supervisado (Independent – Weakly supervised): La base de datos no se hace pública hasta el comienzo de la evaluación. Por un lado, el test se realiza en el entorno del investigador mediante datos sin etiquetar y con limitaciones temporales. Por otro lado, las prestaciones del sistema se determinan por evaluadores independientes a partir de las puntuaciones obtenidas por el investigador durante la prueba del sistema.
- Test independiente y supervisado (Independent – Supervised): La presente aproximación es muy similar a la anterior. Sin embargo, en este caso el test se realiza en la localización del evaluador sobre el hardware del investigador.
- Test independiente y fuertemente supervisado (Independent – Strongly supervised): En este último escenario, la base de datos no se hace pública hasta la conclusión de la evaluación y los algoritmos se evalúan en la ubicación del evaluador sobre su hardware. Los algoritmos se ejecutan en un entorno totalmente controlado, en el que las operaciones de entrada y salida se definen mediante un protocolo a cumplir.

Una evaluación es correcta cuando las puntuaciones que ofrece permiten diferenciar entre las tecnologías y enfoques existentes. Las fortalezas y debilidades encontradas durante un proceso de evaluación indican las aplicaciones para las que las tecnologías son más aptas. A continuación se describen los principales tipos de evaluación [Phillips et al, 2000]:

- Evaluación de tecnología: Es el más general de los tipos de evaluación. Normalmente se realiza sobre algoritmos o prototipos de laboratorio para determinar el Estado del Arte e identificar los progresos tecnológicos.
- Evaluación de escenarios: Valora las prestaciones generales de un sistema para un prototipo de escenario que modela un dominio de aplicación concreto. El principal objetivo de este tipo de evaluación es determinar si una tecnología biométrica es lo suficientemente madura para alcanzar los requisitos de prestaciones en un tipo de aplicaciones. Uno de sus objetivos es determinar las prestaciones de diferentes combinaciones de sensores y algoritmos.
- Evaluación operacional: Es similar a la evaluación de escenarios. Mientras que la evaluación de un escenario valora un determinado tipo de aplicación, la operacional mide las prestaciones de un algoritmo específico en una aplicación concreta. El principal objetivo de una evaluación operacional es determinar si un sistema biométrico alcanza los requisitos de prestaciones de una aplicación específica.

2.5.3. Medidas de Rendimiento

Existen diferentes indicadores que permiten evaluar el rendimiento de los sistemas biométricos. Es posible obtener medidas de prestaciones en cualquiera de las tres operaciones del proceso biométrico (véase la sección 2.2.2).

En la operación de alistamiento pueden aparecer dificultades en la generación de modelos debido a problemas en la adquisición de los datos biométricos [Bolle et al, 2003]. Los siguientes parámetros ofrecen una medida de las prestaciones del proceso de alistamiento [Renesse, 2002]:

- FTA (Failure to Acquire): Porcentaje de usuarios para los que el sistema no es capaz de adquirir muestras utilizables en etapas posteriores.
- FTE (Failure to Enrollment): Porcentaje de usuarios para los que el sistema no es capaz de generar un modelo de suficiente calidad.
- TTE (Time to Enroll): Tiempo transcurrido desde la captura de la muestra a la generación del modelo de usuario.

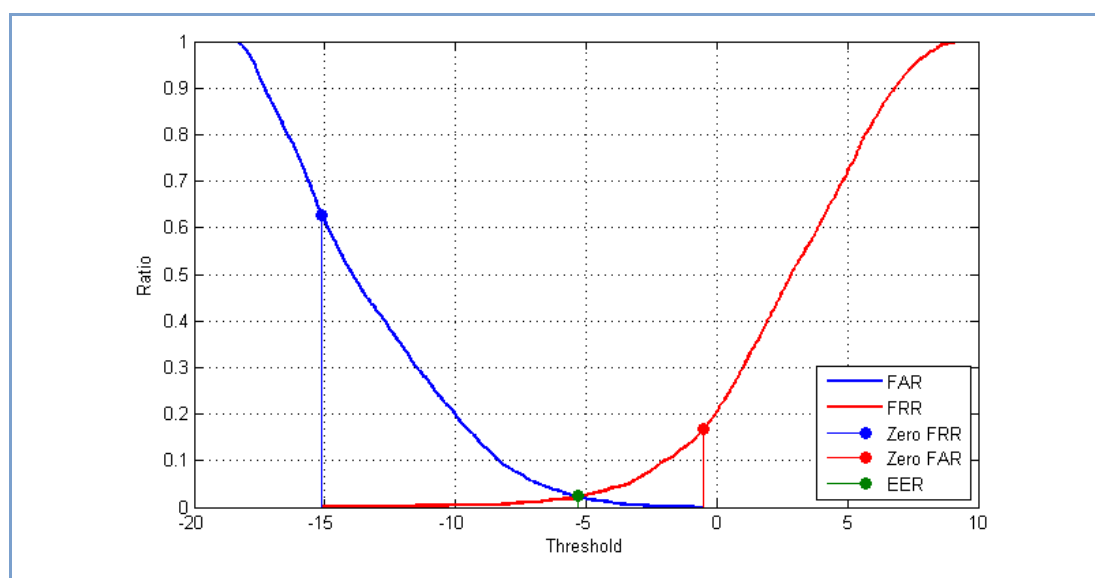


Figura 2.10. Curvas de Falso Rechazo y Falsa Aceptación

La operación de reconocimiento se ve influenciada por factores como las condiciones de adquisición, la variabilidad del rasgo biométrico o la calidad del modelo de usuario [Prabhakar et al, 2003]. Debido a estos factores se pueden producir errores a la hora de verificar o determinar la identidad de un sujeto. En este caso, las principales medidas de calidad son las siguientes:

- FRR (False Rejection Rate): Probabilidad de que el sistema rechace a un usuario legítimo porque no es capaz de identificarlo correctamente.
- FAR (False Acceptance Rate): Probabilidad de que el sistema autentique correctamente a un usuario ilegítimo.

- TTM (Time to Match): Tiempo transcurrido desde el final de la captura de la muestra a la decisión del sistema.

Para obtener un umbral de decisión óptimo, se puede representar el comportamiento de los ratios de falsa aceptación y falso rechazo en función del umbral, tal y como muestra la Figura 2.10.

Existen otras medidas que pueden ser usadas como una medida general de las prestaciones de un sistema biométrico:

- EER (Equal Error Rate): Hace referencia al valor del umbral de decisión para el que los valores de FAR y FRR son iguales.
- HTER (Half Total Error Rate): Es la semisuma de los valores de FAR y FRR.
- Coste: Es la suma ponderada de los valores de FAR y FRR. En este cálculo se asignan costes a los diferentes tipos de errores del sistema.

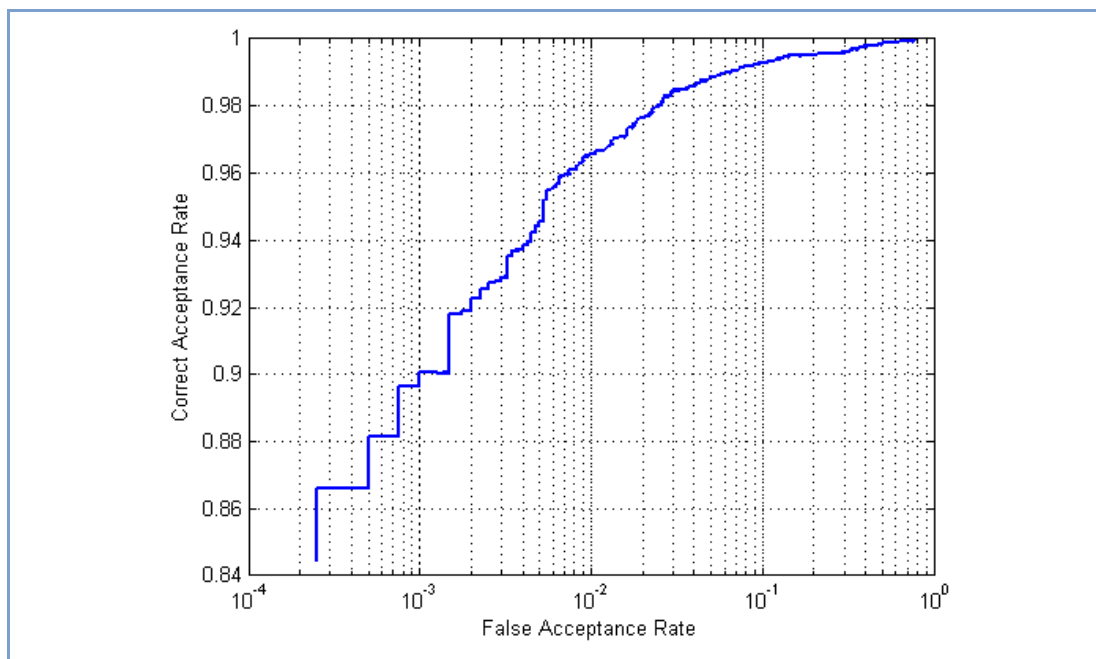


Figura 2.11. Curva ROC

Una manera alternativa de ofrecer las prestaciones de un sistema biométrico es mediante la curva ROC (Receiver Operating Curve) [Fogarty et al, 2005]. Esta curva consiste en la representación gráfica del ratio de falsa aceptación (FAR) frente al ratio de correcta aceptación (1-FRR) en función del umbral de decisión. La Figura 2.11 muestra la curva ROC obtenida a partir de los datos de la Figura 2.10. Una de las principales ventajas de esta curva es la posibilidad de contrastar las prestaciones de diferentes sistemas en un solo gráfico.

METAMODELO

El objetivo del capítulo es ofrecer el metamodelo que defina la sintaxis, semántica e interpretación con las que construir las secuencias de operaciones que constituyen un experimento.

3.1. INTRODUCCIÓN

Un modelo es un conjunto de afirmaciones acerca de una realidad particular. Se emplean como herramientas descriptivas, pero alternativamente, se utilizan para especificar de manera formal sistemas cuyo objetivo es la generación de nuevos modelos [Seidewitz, 2003]. Su aplicación en el desarrollo de aplicaciones no requiere el uso de ningún lenguaje de programación específico para representar los detalles del sistema; simplemente se ha de describir su funcionalidad. Este hecho mejora la productividad ya que se aíslan los elementos que no son de interés [Atkinson et al, 2003].

Un metamodelo es una herramienta de especificación de modelos y es por sí mismo el modelo de la descripción de un dominio particular. La infraestructura necesaria para su creación tiene que definir los siguientes conceptos:

- Interpretación: Criterios que relacionan las afirmaciones contenidas en los modelos con los hechos del mundo real.
- Semántica: Conceptos necesarios para la creación de los modelos.
- Sintaxis: Notación para construir los modelos.
- Transformaciones: Elementos que permiten la correspondencia entre modelos de diferentes ámbitos.

Es importante describir las aplicaciones independientemente de su implementación y puesto que éstas son dinámicas, su formalización conceptual ha de representar su comportamiento [Saake et al, 1993]. Los elementos básicos del modelado conceptual de aplicaciones independientes de la situación son los siguientes:

- Abstracción de los detalles de la implementación tanto en estructuras de datos como en funcionalidades computacionales.

- Semántica formal para conseguir una verificación consistente de la implementación.
- Marco formal para modelar la estructura y el comportamiento.
- Integración de funciones heterogéneas con las que construir sistemas.
- Formalización de las conexiones existentes entre los diferentes componentes.

3.2. OBJETIVO

Tal y como se ha presentado en la sección 2.2, el proceso biométrico puede ser descrito como una serie de operaciones sobre un flujo lineal de datos. Un experimento pretende simular este proceso en un entorno controlado con el objetivo de valorar los algoritmos utilizados. En otras palabras, es una sucesión de operaciones que transforman una información biométrica en otra con el fin de obtener las prestaciones de los propios procesos. Conforme a esto, un experimento se fundamenta en dos elementos básicos: datos y operaciones.

Los datos engloban a cualquier tipo de información que se utiliza o genera durante la ejecución de un experimento. A medida que avanza la ejecución, éstos se vuelven menos voluminosos y más específicos. De forma más concreta, cualquier tipo de información presente en un experimento se puede clasificar como:

- **Muestra:** Información obtenida directamente de los rasgos fisiológicos o comportamentales de un sujeto. Un fragmento de voz o la captura de una imagen de iris son ejemplos de datos pertenecientes a este tipo.
- **Vector de Características:** Datos obtenidos a partir de la transformación matemática de las muestras biométricas de un individuo.
- **Modelo:** Conjunto de datos que definen el tipo o la identidad de un usuario.
- **Puntuación:** Valoración numérica que asigna un sistema de comparación de patrones a un conjunto de características biométricas.
- **Decisión:** Evaluación realizada por un sistema de clasificación acerca de la identidad de un sujeto.
- **Resultado:** Datos que contienen medidas de prestaciones para un sistema. A este tipo pertenecen las curvas ROC o las matrices de confusión.

Conceptualmente, los datos pueden ser agrupados en flujos con el fin de modelar su transporte entre las distintas operaciones. La información biométrica de los flujos puede ser estructurada como conjuntos de vectores de

datos.

Las operaciones representan a todos aquellos procedimientos que manipulan datos dentro de un experimento. Es posible diferenciar dos tipos de operaciones:

- De control: Procedimientos que manipulan flujos de datos para modificar su estructura.
- De transformación: Operaciones que procesan información mediante la aplicación de funciones o algoritmos matemáticos.

Las operaciones son elementos que generan datos a partir de los recibidos de uno o varios flujos. Por este motivo, pueden ser asimiladas como nodos dentro de un experimento.

La Figura 3.1 muestra el diagrama esquemático del entorno de evaluación. A partir de una base de datos biométrica y un protocolo de pruebas se puede construir un modelo para definir la ejecución un experimento. Su objetivo es simular el comportamiento del sistema biométrico y evaluar los algoritmos empleados.

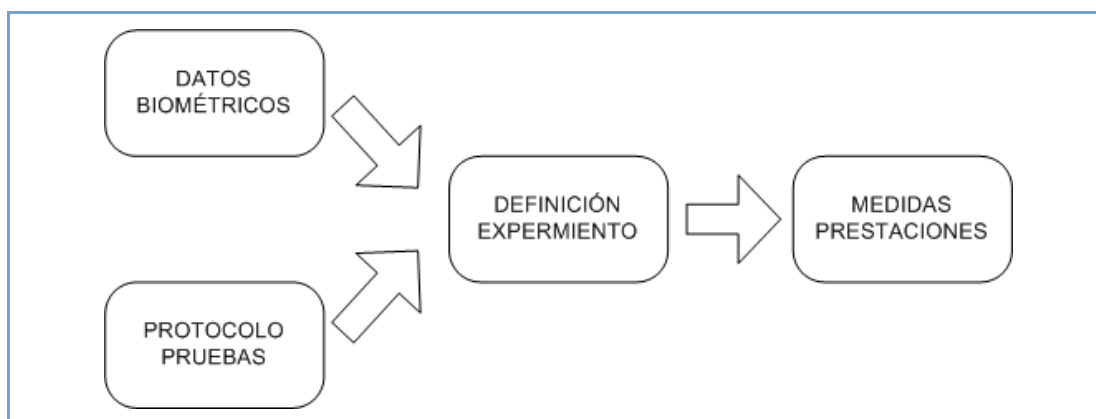


Figura 3.1. Marco de Evaluación de un Sistema Biométrico

El objetivo del metamodelo propuesto es establecer las bases para la descripción de los datos y las operaciones que componen los experimentos de las evaluaciones de tecnología off-line (véase la sección 2.5.2). Las bases del proceso de experimentación tienen que establecer métodos capaces de describir las etapas de extracción de características, comparación de patrones, generación de modelos y en general, todas aquellas que forman parte de las operaciones de alistamiento, verificación y reconocimiento (véase sección 2.2.2). Puesto que el objetivo es modelar la evaluación off-line, no se pretende reproducir la adquisición de los datos biométricos (véase sección 2.2.1).

3.3. CONTEXTO ALGEBRAICO

Si se analiza un experimento biométrico desde el punto de vista algebraico, es

posible describirlo como una secuencia de operaciones sobre un conjunto de vectores de datos. Un vector es un conjunto ordenado de datos que puede contener valores de tipo escalar, texto, tuplas, conjuntos de vectores, etc.

Un experimento biométrico no es más que la transformación de un conjunto de vectores de datos en otro conjunto de vectores de resultados [Puente, 2007]. En su contexto, Σ determina el espacio de vectores que se pueden formar de manera homogénea, Ω define un conjunto cualquiera de vectores y por último, Δ establece el dominio de un espacio. La Figura 3.2 recoge la descripción algebraica de un experimento mediante la notación propuesta.

$$\begin{aligned}
 &f: \Delta_{\mathbb{X}} \rightarrow \Delta_{\mathbb{Y}} \mid f(\Omega_{\mathbb{X}}) = \Omega_{\mathbb{Y}} \\
 &\Omega_{\mathbb{Y}} = \{y = f(x) \mid x \in \Omega_{\mathbb{X}}\} \\
 &\Sigma_{\mathbb{X}} \supset \Delta_{\mathbb{X}} \ni x = \{x_1, \dots, x_m\} \\
 &\Sigma_{\mathbb{Y}} \supset \Delta_{\mathbb{Y}} \ni y = \{y_1, \dots, y_n\}
 \end{aligned}$$

Figura 3.2. Definición Algebraica de un Experimento

La transformación de los datos durante un experimento se puede dividir mediante descomposición en operaciones básicas [Puente, 2007]. Estas operaciones son procesos de alto nivel y por lo tanto, pueden ser descritas mediante una serie de primitivas con el fin de soportarlas.

3.3.1. Identidad (\mathbb{I})

La transformación primitiva \mathbb{I} obtiene un conjunto de vectores idéntico al dado asignando a cada vector x de $\Omega_{\mathbb{X}}$ su mismo vector x .

$$\begin{aligned}
 &\mathbb{I}: \Delta_{\mathbb{X}} \rightarrow \Delta_{\mathbb{X}} \mid \mathbb{I}(\Omega_{\mathbb{X}}) = \Omega'_{\mathbb{X}} \\
 &\Omega'_{\mathbb{X}} = \{x' = x \mid x \in \Omega_{\mathbb{X}}\}
 \end{aligned}$$

Figura 3.3. Primitiva Identidad

3.3.2. Proyección (π)

La transformación primitiva π_c obtiene el conjunto de vectores $\Omega_{\mathbb{Y}}$ a partir de la reducción dimensional del conjunto $\Omega_{\mathbb{X}}$ según los índices de los campos especificados en el conjunto S .

$$\begin{aligned}
 \pi_S: \Delta_X &\rightarrow \Delta_Y \mid \pi_S(\Omega_X) = \Omega_Y \\
 S &= \{a_1, \dots, a_m\}, a_i \in \mathbb{N} \\
 x &= \{x_1, \dots, x_n\}, y = \{y_1, \dots, y_m\} \\
 \Omega_Y &= \{y \mid \forall x \in \Omega_X \exists y \in \Omega_Y \mid y_i = x_{a_i} \forall 1 \leq i \leq m\}
 \end{aligned}$$

Figura 3.4. Primitiva Proyección.

3.3.3. Selección (σ)

La transformación primitiva σ_C obtiene el subconjunto Ω' a partir de los vectores del conjunto Ω que cumplen la condición C dada.

$$\begin{aligned}
 \sigma_C: \Delta_X &\rightarrow \Delta_X \mid \sigma_C(\Omega_X) = \Omega'_X \\
 C: \Sigma_X &\rightarrow \{\text{false}, \text{true}\} \\
 \Omega'_X &= \{x' = x \mid x \in \Omega_X \mid C(x) = \text{true}\}
 \end{aligned}$$

Figura 3.5. Primitiva Selección.

3.3.4. Asociación (\mathbb{J})

La transformación primitiva \mathbb{J}_C genera un nuevo conjunto de vectores Ω_Y a partir de N conjuntos Ω_{X_i} . Los vectores de Ω_Y se obtienen como la concatenación de aquellos vectores de los conjuntos de partida que cumplen la condición C dada.

$$\begin{aligned}
 \mathbb{J}_C: \Delta_{X_1}, \dots, \Delta_{X_N} &\rightarrow \Delta_Y \mid \mathbb{J}_C(\Omega_{X_1}, \dots, \Omega_{X_N}) = \Omega_Y \\
 \Omega_Y &= \left\{ y = \bigcup_i x_i \mid \forall x_i \in \Omega_{X_i} \mid C(x_1, \dots, x_N) = \text{true} \right\} \\
 C: \Sigma_{X_1}, \dots, \Sigma_{X_N} &\rightarrow \{\text{false}, \text{true}\}
 \end{aligned}$$

Figura 3.6. Primitiva Asociación.

3.3.5. Agrupación (\mathbb{G})

La transformación primitiva \mathbb{G}_C permite obtener un recubrimiento del área de datos según una determinada condición C .

$$\begin{aligned}
 &\mathbb{G}_C: \Delta_{\mathbb{X}} \rightarrow \Delta_{\mathbb{Y}} \mid \mathbb{G}_C(\Omega_{\mathbb{X}}) = \Omega_{\mathbb{Y}} \\
 &\mathcal{C}: \Sigma_{\mathbb{X}}, \mathbb{N} \rightarrow \{\text{false}, \text{true}\} \\
 &\Omega_{\mathbb{Y}} = \left\{ \Omega_{\mathbb{X}_i} \mid \bigcup_i \Omega_{\mathbb{X}_i} = \Omega_{\mathbb{X}} \wedge \Omega_{\mathbb{X}_i} \cap \Omega_{\mathbb{X}_j} = \emptyset \forall i \neq j \right\} \\
 &\Omega_{\mathbb{X}_i} = \{x \mid \mathcal{C}(x, i) = \text{true}\} \\
 &\forall x \in \Omega_{\mathbb{X}} \exists i \mid x \in \Omega_{\mathbb{X}_i}
 \end{aligned}$$

Figura 3.7. Primitiva Agrupación.

3.3.6. Procesado (\mathbb{P})

La transformación primitiva \mathbb{P}_f genera un conjunto $\Omega_{\mathbb{Y}}$ a partir del procesado individual, según la función f , de cada uno de los vectores del conjunto $\Omega_{\mathbb{X}}$. Cada uno de los vectores de salida contiene los elementos de los vectores de partida y además, incorpora los nuevos elementos generados por la función.

$$\begin{aligned}
 &\mathbb{P}_f: \Delta_{\mathbb{X}} \rightarrow \Delta_{\mathbb{Y}} \mid \mathbb{P}_f(\Omega_{\mathbb{X}}) = \Omega_{\mathbb{Y}} \\
 &\Omega_{\mathbb{Y}} = \{y = f(x) \mid \forall x \in \Omega_{\mathbb{X}}\}
 \end{aligned}$$

Figura 3.8. Primitiva Proceso.

3.4. DEFINICIÓN

El metamodelo persigue la sencillez en las definiciones de los experimentos a través de la notación gráfica. Un mismo modelo consta de dos vistas: una gráfica y otra formal.

La representación gráfica de un experimento se denomina EPC (Experiment Process Chart). Su principal objetivo es simplificar la construcción y edición de modelos. EPC se centra en la descripción de la cadena de procesos que suceden durante la ejecución y se basa en los diagramas de flujo o actividad de UML (Unified Modeling Language) [OMG, 2009].

La representación detallada se designa mediante el acrónimo BED (Biometric Experiment Definition). El objetivo de ésta es ofrecer la descripción completa de las operaciones y los datos implicados en la ejecución de un experimento y se fundamenta en el lenguaje XML (eXtended Markup Language) [W3C, 2004].

Tal y como se ha presentado en la sección 3.1, es necesario definir una serie

de conceptos con el objetivo de obtener un metamodelo consistente. En las siguientes páginas se desarrolla la interpretación, la semántica y la sintaxis.

3.4.1. Interpretación

De manera introductoria, en el modelo de ejemplo de la Figura 3.9 se representa el proceso de entrenamiento y verificación de la identidad para uno o varios usuarios.

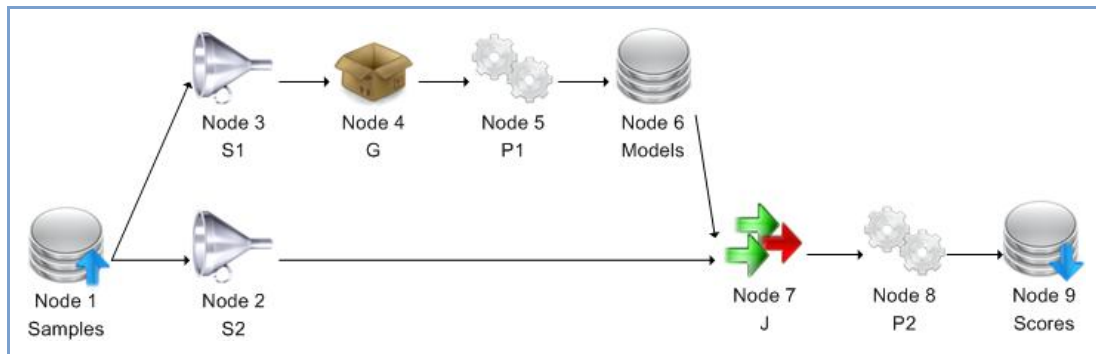


Figura 3.9. Ejemplo del Modelo de un Experimento

Este experimento no es más que la transformación de un conjunto de muestras biométricas (“Samples”) en un conjunto de puntuaciones (“Scores”) y puede ser descrito a través de los siguientes pasos:

1. Se parte de las muestras almacenadas de cierta característica biométrica en forma de vectores de datos (Nodo 1 “Samples”).
2. De todos los vectores almacenados, se seleccionan unas muestras para la etapa de entrenamiento y otras para el proceso de identificación (Nodo 2 “S₂” y Nodo 3 “S₁”).
3. Los vectores de entrenamiento son agrupados (Nodo 4 “G”), procesados mediante un algoritmo de generación de modelos (Nodo 5 “P₁”) y almacenados (Nodo 6 “Models”).
4. Se fusionan los vectores de características seleccionados por el nodo 2 con los modelos generados (Nodo 7 “J”).
5. Los vectores de características se procesan junto con los modelos mediante un algoritmo de clasificación de patrones (Nodo 8 “P₂”).
6. Las puntuaciones generadas por el algoritmo son almacenadas (Nodo 9 “Scores”).

El metamodelo se define de acuerdo al modelo algebraico presentado en la sección 3.3 con el objetivo de dar soporte conceptual a las operaciones presentadas en el ejemplo anterior. En la Figura 3.10 se presenta la descripción formal del metamodelo mediante un diagrama de clases. Cada una de las operaciones definidas lleva asociado un conjunto de puntos de entrada y

de salida, cuya cardinalidad depende de la propia operación.

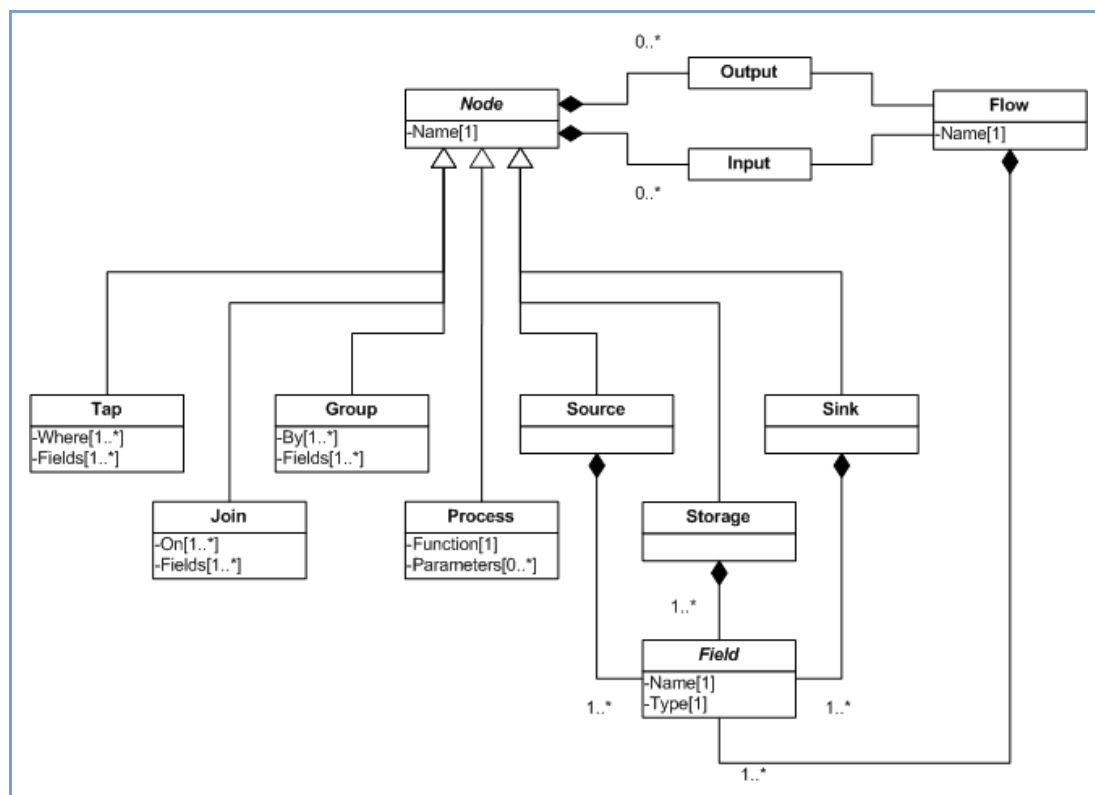


Figura 3.10. Diagrama de Clases del Metamodelo

3.4.2. Semántica

El metamodelo propuesto se orienta hacia la descripción de la ejecución de un experimento mediante la sucesión de operaciones sobre flujos de vectores. En el modelo de un experimento se pueden distinguir dos tipos de elementos. Por un lado, los nodos (asociados a las operaciones) establecen puntos para el control, procesamiento y almacenamiento de la información y por otro, los flujos (relacionados con los datos) definen caminos para la transferencia de vectores entre dos nodos.

3.4.3. Sintaxis

Finalmente se presenta la sintaxis propuesta para describir los elementos del metamodelo.

3.4.3.1. Flujo

Un flujo es un elemento que modela el transporte de datos desde la salida de un nodo hacia la entrada de cualquier otro. Un flujo queda caracterizado por:

- El nodo origen que inyecta vectores.
- El nodo destino que los recibe.

- La estructura del flujo y los propios datos.

3.4.3.2. Nodos

Un nodo es un elemento que modela la generación, el control, el procesado o el almacenamiento de datos en un experimento. A continuación se ofrece la descripción de cada uno de los nodos definidos en el metamodelo. La Figura 3.11 ofrece los elementos visuales asociados a las operaciones.

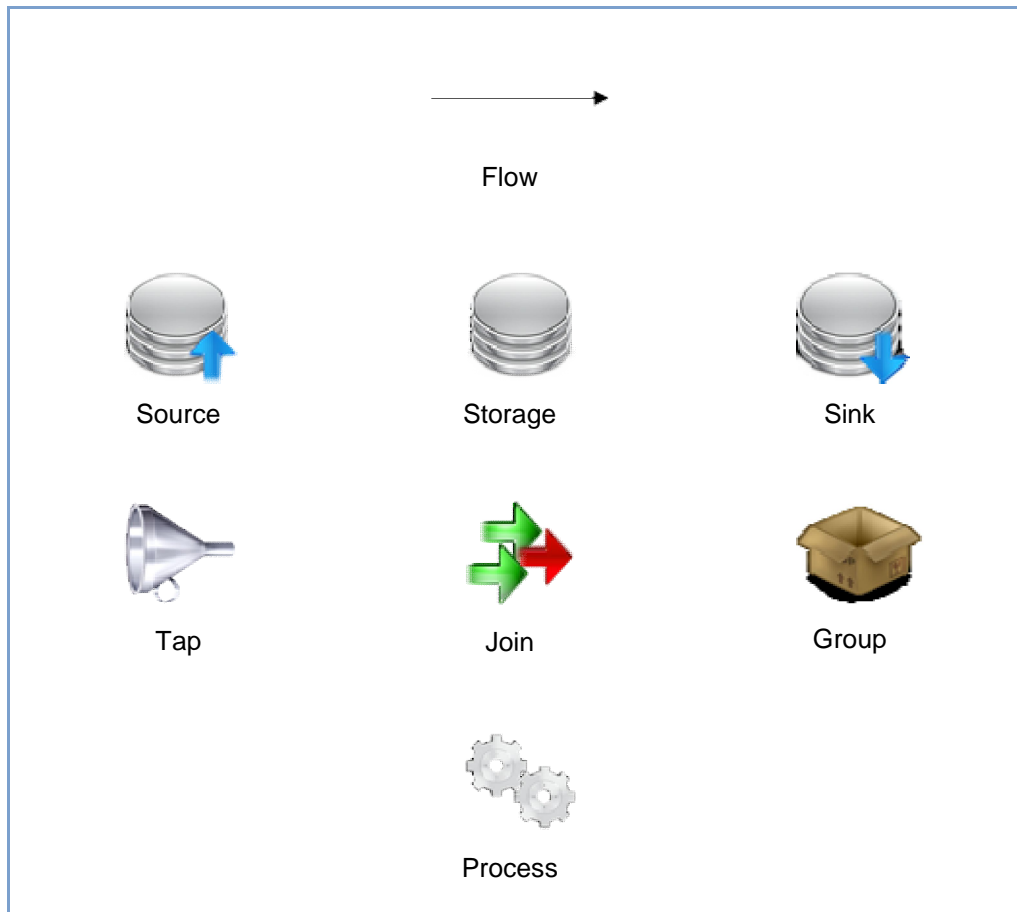


Figura 3.11. Elementos Visuales Asociados a las Operaciones

Nodos Source, Storage y Sink

Estos nodos modelan las propiedades tanto de las bases de datos que se emplean para evaluar un sistema como de la información biométrica presente en el experimento. Los almacenes de información biométrica representan la abstracción del lugar físico donde se almacenan vectores homogéneos. Un almacén de información no es más que una tabla desde la que se extraen o insertan datos durante la ejecución de un experimento.

El nodo Source establece la obtención de datos desde un almacén de información. Este nodo solamente posee flujos de salida puesto que es un punto de inicio en la ejecución. Así mismo, dentro de un mismo modelo pueden aparecer varias fuentes de datos representadas por diferentes nodos de tipo Source.

El nodo Sink modela el volcado de datos hacia un almacén de información. Como su propio nombre indica, es un punto final en la ejecución de un experimento y por lo tanto, no posee flujos de salida.

El nodo Storage define el almacenamiento de información temporal durante un experimento. El nodo presenta un único flujo de entrada, cuyo contenido se almacena hasta el final de la ejecución, y uno o más flujos de salida en los que se replica su contenido. Así mismo, ofrece la posibilidad de volcar el flujo de entrada en un almacén de información tras la ejecución.

Nodo Tap

El nodo Tap modela la aplicación de un filtro sobre un flujo de datos y por lo tanto, posee una entrada y una salida. Esta última contiene aquellos vectores que cumplen los criterios de selección definidos en el atributo *where*. Así mismo, permite realizar una proyección de los vectores mediante el atributo *fields*.

$$\Omega_Y = \text{Tap}(\Omega_X) = T(\Omega_X) = \pi_{fields}(\sigma_{where}(\Omega_X))$$

Figura 3.12. Definición Algebraica del Nodo Tap

Nodo Join

El nodo Join establece las propiedades de la asociación de varios flujos. Las relaciones de igualdad o criterios de combinación que se tienen que dar entre los vectores se definen mediante el atributo *On*. Al igual que el nodo Tap, el presente nodo permite definir una proyección de los vectores de salida mediante el atributo *fields*.

$$\begin{aligned}\Omega_Y &= \text{Join}(\Omega_{X_1}, \dots, \Omega_{X_N}) \\ &= J(\Omega_{X_1}, \dots, \Omega_{X_N}) = \pi_{fields}(\mathbb{J}_{On}(\Omega_{X_1}, \dots, \Omega_{X_N}))\end{aligned}$$

Figura 3.13. Definición Algebraica del Nodo Join

Nodo Group

El nodo Group define la agrupación de los vectores de un flujo según unos determinados criterios.

$$\Omega_Y = \text{Group}(\Omega_X) = G(\Omega_X) = \mathbb{G}_{by}(\pi_{fields}(\Omega_X))$$

Figura 3.14. Definición Algebraica del Nodo Group

La agrupación de los vectores se realiza de acuerdo a los valores de los campos definidos por el atributo *by*. El nodo permite agrupar los vectores de un

flujo, según los valores que toman ciertos campos, para generar diferentes subconjuntos a la salida. De nuevo es posible realizar una proyección de los vectores de entrada de acuerdo a los campos listados en el atributo *fields*.

Nodo Process

El nodo Process modela el procesamiento de los vectores contenidos en un flujo mediante la aplicación de algoritmos y funciones. Este nodo es el elemento principal en la definición de cualquier experimento puesto que representa la aportación externa de los algoritmos a evaluar. Mientras que los nodos anteriores se centran en la selección y la agrupación de los datos biométricos, este nodo lo hace en el procesamiento individual de cada uno de los vectores que forman parte de un determinado flujo.

$$\Omega_Y = \text{Process}(\Omega_X) = P(\Omega_X) = \pi_{fields} \left(\mathbb{P}_f(\Omega_X) \right)$$

Figura 3.15. Definición Algebraica del Nodo Process

De forma general el nodo Process posee una entrada y una salida; sin embargo, puede comportarse como una fuente de datos. En estos casos, el nodo no necesita el flujo de entrada.

PLATAFORMA BEEP

Este capítulo ofrece la visión general de la plataforma de experimentación y presenta las restricciones impuestas a su implementación. Así mismo, recoge la definición de los subsistemas y los elementos comunes a ellos.

4.1. INTRODUCCIÓN

BEEP surge como una aplicación web capacitada para ofrecer servicios de experimentación biométrica a los usuarios bajo un entorno común. La plataforma está compuesta por dos subsistemas: el Front-End y el Back-End.

El Front-End es el subsistema encargado de la interacción con el usuario. Ofrece las funcionalidades necesarias para crear y modificar modelos de experimentos y presentar informes de resultados de los sistemas evaluados. Así mismo, el Front-End brinda mecanismos para incorporar descripciones de procesos y de datos biométricos a los modelos.

El Back-End es el subsistema responsable de la ejecución de los experimentos. Contiene las funcionalidades necesarias para interpretar y ejecutar los modelos de los experimentos. De la misma manera, es responsable de la generación de los informes de prestaciones.

El diagrama de la Figura 4.1 muestra la interacción entre los elementos de la plataforma. A continuación se describe, de forma general, la relación entre el usuario y la aplicación:

1. El usuario define el modelo de un experimento al que incorpora las descripciones de datos biométricos y procesos.
2. El Front-End almacena la información relacionada con el experimento en el modelo de datos de la plataforma.
3. El Front-End notifica al Back-End de la solicitud de ejecución del experimento a través de un interface de comunicación.
4. El Back-End recupera del modelo de datos los recursos necesarios para la ejecución.

5. El Back-End ejecuta el experimento y almacena en el modelo de datos los resultados generados.
6. El Front-End accede al modelo de datos para recuperar el informe de prestaciones del sistema evaluado.
7. El usuario obtiene el informe de prestaciones del sistema biométrico evaluado.

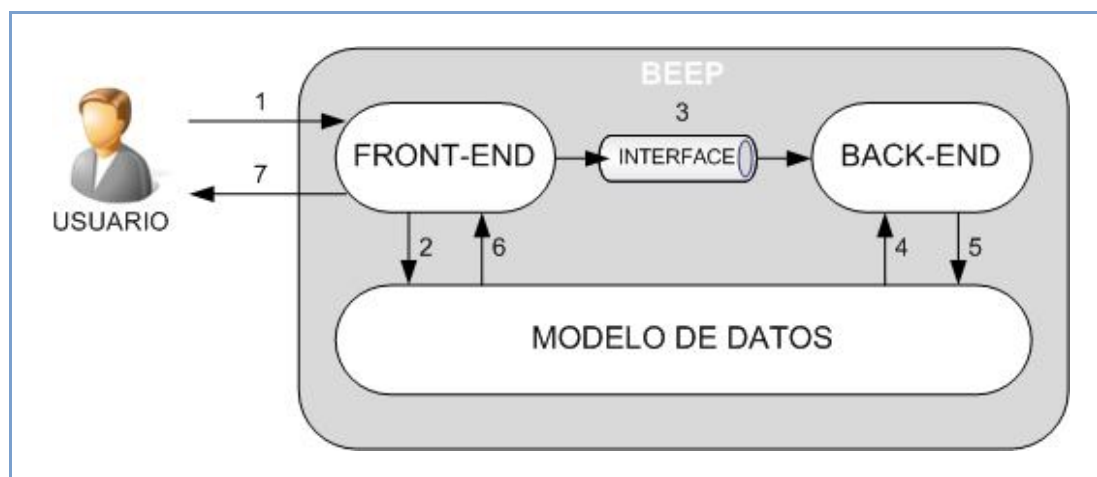


Figura 4.1. Visión General de la Plataforma.

Para dar soporte a los dos subsistemas existe una serie de componentes comunes:

- Modelo de datos: La capa de persistencia alberga los datos de la plataforma y es utilizada por ambos subsistemas (véase la sección 4.4).
- Interface de datos: El metamodelo presentado define el interface de datos entre el Front-End y el Back-End (véase la sección 4.5).
- Interface de comunicación: Establece el protocolo de mensajes entre los dos subsistemas (véase la sección 4.6).

4.2. ESPECIFICACIÓN DE REQUISITOS

En las siguientes páginas se ofrece la especificación de requisitos de la plataforma. Ésta recoge los requisitos de capacidad y de restricción para la plataforma y establece el punto de partida para el análisis funcional del Front-End (véase el Capítulo 5).

4.2.1. Requisitos de Capacidad

Los requisitos de capacidad son aquellas funciones requeridas por los usuarios para resolver un problema o alcanzar un objetivo [ESA, 1991]. La Tabla 4.1 enumera cada uno de estos requisitos (denominados RC).

ID	ESENCIAL	MODIFICABLE	PRIORIDAD
RC-001	Si	No	Alta
DESCRIPCIÓN: La plataforma ofrecerá servicios con independencia de la ubicación del usuario.			
RC-002	Si	No	Alta
DESCRIPCIÓN: La plataforma ofrecerá mecanismos para crear y modificar modelos de experimentos biométricos			
RC-003	Si	Si	Alta
DESCRIPCIÓN: La plataforma ofrecerá mecanismos para modelar datos biométricos con independencia de su origen.			
RC-004	Si	No	Alta
DESCRIPCIÓN: La plataforma tendrá capacidad para asegurar la información biométrica almacenada.			
RC-005	Si	No	Alta
DESCRIPCIÓN: La plataforma ofrecerá mecanismos para modelar algoritmos de tratamiento de información biométrica.			
RC-006	Si	No	Alta
DESCRIPCIÓN: La plataforma tendrá capacidad para interpretar y ejecutar modelos de experimentos biométricos.			
RC-007	Si	Si	Alta
DESCRIPCIÓN: La plataforma tendrá capacidad para generar y presentar informes de prestaciones tras la ejecución de los experimentos.			

Tabla 4.1. Requisitos de Capacidad de la Plataforma

4.2.2. Requisitos de Restricción

Los requisitos de restricción engloban a las limitaciones impuestas por los usuarios sobre la forma en la que se resuelve un problema o alcanza un objetivo [ESA, 1991]. En la Tabla 4.2 se muestran las restricciones de la plataforma (denominadas RP) impuestas por el cliente.

ID	ESENCIAL	MODIFICABLE	PRIORIDAD
RP-001	Si	No	Alta
DESCRIPCIÓN: La plataforma establece su arquitectura física como cliente-servidor.			
RP-002	Si	No	Alta
DESCRIPCIÓN: El modelo de datos de la plataforma tiene que ser común para los dos subsistemas.			
RP-003	Si	Si	Alta
DESCRIPCIÓN: El interface de datos entre los dos subsistemas queda establecido por la definición de los modelos de los experimentos.			
RP-004	Si	No	Alta
DESCRIPCIÓN: Es necesario un protocolo de comunicación entre los subsistemas.			

Tabla 4.2. Requisitos de Restricción de la Plataforma

4.3. ARQUITECTURA

La división física de la plataforma se asienta en una arquitectura cliente-servidor tal y como muestra la Figura 4.2:

- Cliente: Máquina que inicia peticiones de servicio. La petición inicial puede convertirse en múltiples requerimientos de trabajo a través de peticiones basadas en HTTP (Hyper Text Transfer Protocol).
- Servidor: Máquina que ofrece los recursos de cómputo dedicados a

responder a las peticiones del cliente. El servidor está conectado a Internet con el objetivo de proveer servicios de forma simultánea a los clientes.

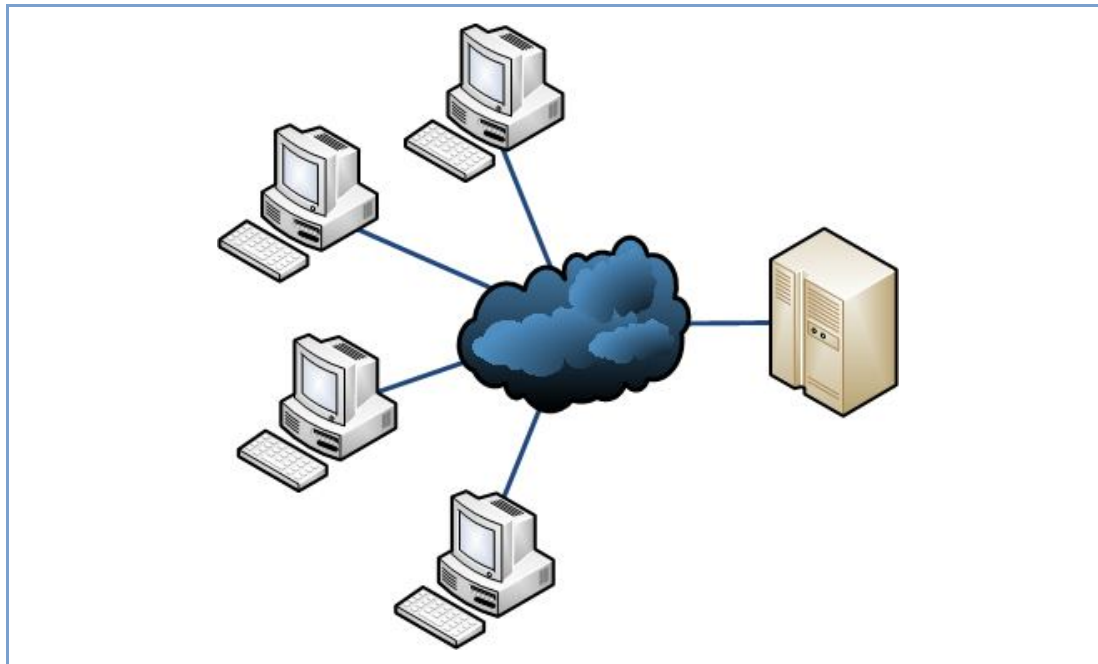


Figura 4.2. Arquitectura Física de la Plataforma

La infraestructura hardware de la máquina servidor se basa en un equipo HP Proliant 115. Éste incorpora un procesador AMD Opteron Dual a 1.8 GHz, 4 GB de memoria RAM y un disco duro de 160 GB. El equipo se encuentra ubicado en un entorno estable con las medidas de seguridad adecuadas con el fin de garantizar la disponibilidad y la integridad de los servicios de la aplicación. Por su parte, la infraestructura software del servidor se basa en Microsoft Windows Server 2003⁵. Éste es un sistema operativo de propósito múltiple capaz de manejar una amplia gama de funciones de servidor, tanto de manera centralizada como distribuida.

El diagrama de despliegue mostrado en la Figura 4.3 ofrece las relaciones entre los componentes lógicos desplegados sobre la arquitectura física impuesta.

La arquitectura lógica de la máquina cliente se compone de un navegador web y de la aplicación de edición de experimentos. Las funciones principales de estos elementos son el acceso a la aplicación web y la construcción de los modelos respectivamente.

Dentro de la arquitectura lógica del servidor aparecen el Front-End y el Back-End. Así mismo, se encuentran las capas de infraestructura y persistencia. La infraestructura de la plataforma se basa en MySQL Connector para acceder a los datos de la plataforma y ODBC para aislar los orígenes de los datos

⁵ <http://www.microsoft.com/spain/windowsserver2003/default.msp>

biométricos de la lógica de la aplicación. Finalmente, la persistencia se consigue mediante sistemas de bases de datos como MySQL, SQL Server y Microsoft Access.

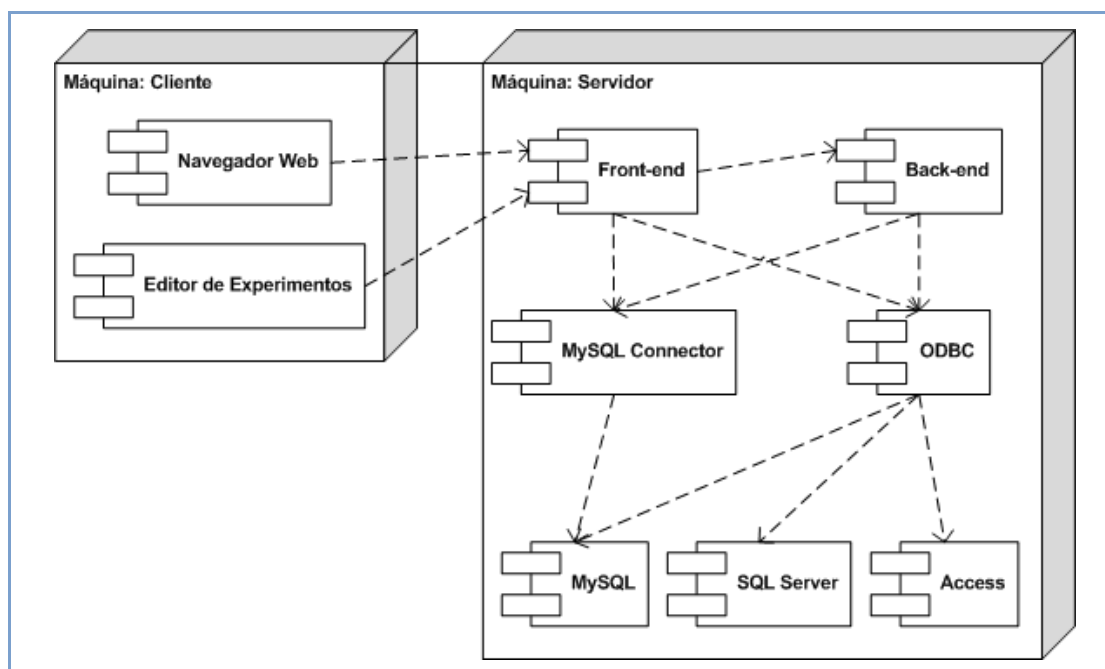


Figura 4.3. Arquitectura Lógica de la Plataforma

4.4. MODELO DE DATOS

En la actualidad, Internet se está convirtiendo en una plataforma de computación y en un interface común para compartir datos [Atzeni et al, 1998]. Dentro de cualquier organización son necesarias tareas como la gestión y la transmisión de información y varios de los problemas que se presentan a la hora de realizar estas tareas son las redundancias y las incongruencias en los datos almacenados.

Durante la etapa de diseño, una base de datos ha de pasar por el proceso de normalización. Existen diferentes reglas de normalización y cada una de ellas se denomina forma normal [Codd, 1970]. Aunque existen diferentes niveles de normalización, la tercera forma normal se considera el máximo nivel necesario para la mayor parte de las aplicaciones [MSDN, 2007].

El modelo E-R es un conjunto de conceptos y reglas para representar de forma global los aspectos lógicos de los diferentes tipos de datos existentes en un sistema [Piattini et al, 2006].

El modelo de datos de la plataforma se ha implementado mediante diferentes bases de datos. Por un lado, se encuentra la base de datos del sistema y por otro, se encuentran las bases de datos de las entidades de investigación.

4.4.1. Modelo de Datos del Sistema

La base de datos del sistema contiene la información de las entidades de investigación, de la cola de ejecución y de los procesos y almacenes de información públicos.

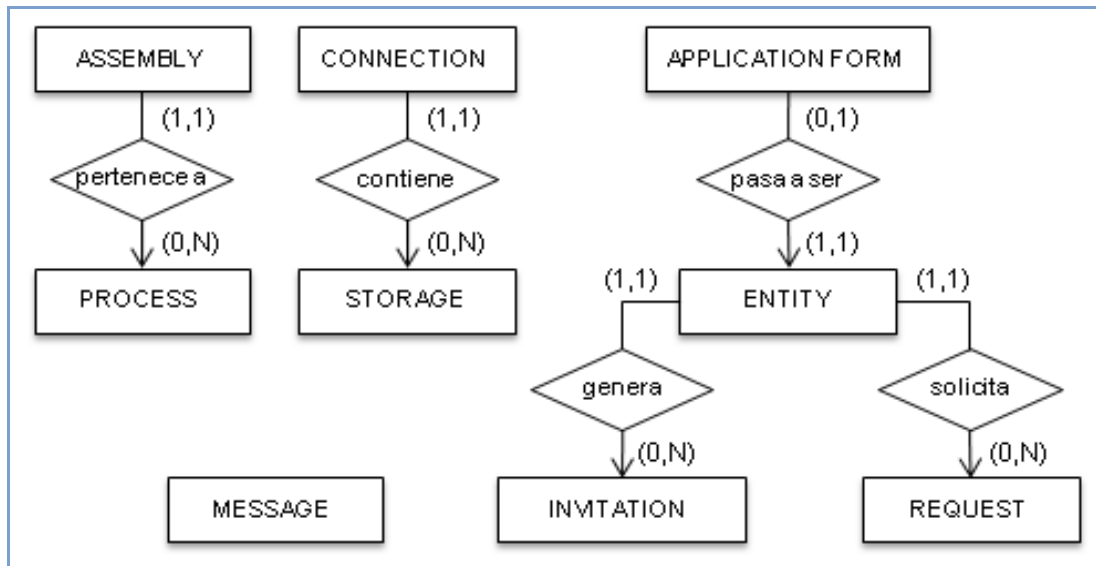


Figura 4.4. Diagrama E-R para el Modelo de Datos del Sistema.

El modelo E-R propuesto para la base de datos del sistema, cuya representación recoge la Figura 4.4, está formado por un total de nueve entidades conceptuales.

- Entidad Application Form: La entidad Application Form representa la solicitud de creación de una entidad investigadora.
- Entidad Entity: En esta entidad se engloba el concepto de entidad u organización investigadora, formada por usuarios y grupos de trabajo, cuyos objetivos son la realización de experimentos y la obtención de resultados.
- Entidad Invitation: La entidad Invitation engloba las invitaciones de creación de nuevos usuarios de la plataforma.
- Entidad Request: En la presente entidad se engloba el concepto de solicitud de ejecución de un experimento por parte de una entidad investigadora.
- Entidad Assembly: La entidad Assembly representa a las librerías de código que necesita un proceso público para realizar sus operaciones
- Entidad Process: La entidad Process abarca a todos aquellos elementos que intervienen en la ejecución de un experimento modificando o tratando los datos biométricos. La presente entidad modela los procesos ofrecidos por la plataforma y que por lo tanto, son de carácter público.

- Entidad Connection: La entidad Connection representa la localización de un almacén de información biométrica.
- Entidad Storage: La entidad Storage recoge el concepto de almacén público de información biométrica.
- Entidad Message: La última entidad del modelo representa a los mensajes públicos de la plataforma.

Dentro del presente modelo de datos, se han definido un total de cinco relaciones:

- Relación Application Form – pasa a ser – Entity: Esta primera relación establece la correspondencia entre las solicitudes de creación y las entidades investigadoras. La relación Application Form – pasa a ser – Entity posee una cardinalidad (0,1) ya que la creación de la entidad depende de la valoración de la solicitud de creación. En el camino inverso, se puede apreciar que la cardinalidad de la relación Entity – procede de – Application Form es (1,1) ya que una entidad investigadora solamente proviene de una solicitud de creación.
- Relación Entity – genera – Invitation: Esta relación presenta la dependencia existente entre las invitaciones de nuevos usuarios y las entidades de investigación. Una entidad de investigación puede generar cero o más invitaciones, por lo que la cardinalidad de la relación es (0,N). En el camino inverso, la relación Invitation – pertenece a – Entity posee una cardinalidad (1,1) ya que una invitación pertenece solamente a una entidad de investigación.
- Relación Entity – solicita – Request: Mediante la presente relación se asocian las entidades de investigación y las solicitudes de ejecución de experimentos. Una entidad puede poseer cero o más solicitudes de ejecución, por lo que la cardinalidad de la relación es (0,N). Si se comprueba la cardinalidad en el camino inverso, la relación Request – pertenece a – Entity posee una cardinalidad (1,1) puesto que una solicitud de ejecución pertenece a una única entidad investigadora.
- Relación Connection – contiene - Storage: Mediante esta relación se asocia cada uno de los almacenes de información con su localización física. La cardinalidad de la relación es (0,N) ya que una conexión puede no contener ningún almacén de información. La relación inversa Storage – pertenece a – Connection posee una cardinalidad (1,1) ya que un almacén de información solamente se localiza mediante una conexión.
- Relación Assembly – pertenece a – Process: La presente relación asocia los procesos biométricos y las librerías de código necesario para ejecutarlos. La cardinalidad de la relación es (0,N) puesto que una librería puede no usarse en ningún proceso. En la relación inversa Process – necesita – Assembly, la cardinalidad de la relación es (1,N) debido a que un proceso necesita como mínimo una librería de código.

4.4.2. Modelo de Datos de la Entidad Investigadora

La base de datos de la entidad de investigación contiene la información de sus usuarios y grupos de trabajo, así como la información de sus experimentos y procesos y almacenes privados.

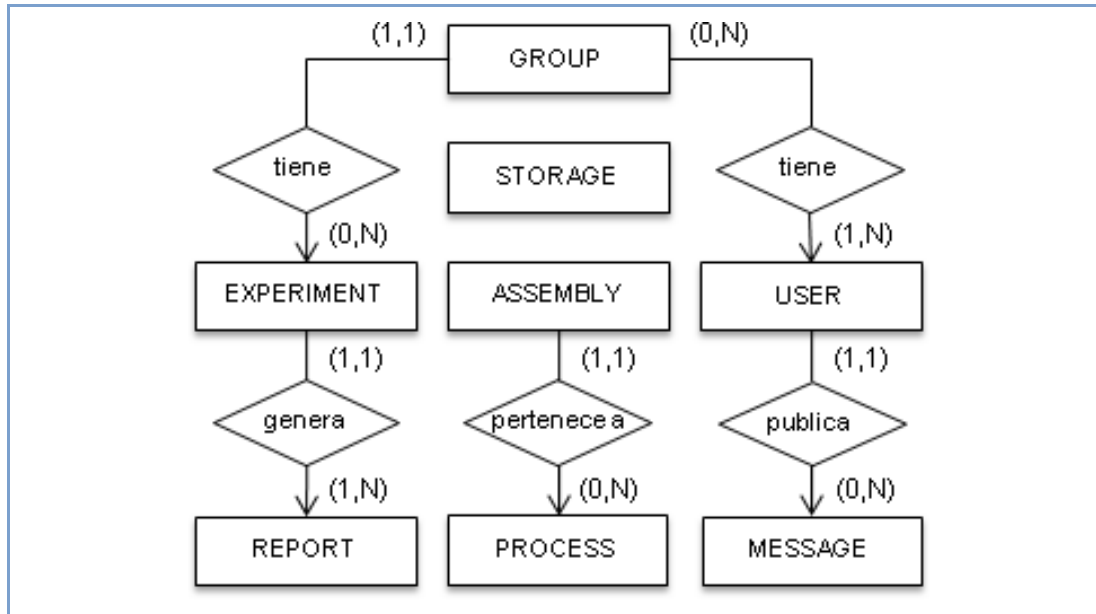


Figura 4.5. Diagrama E-R para el Modelo de Datos de la Entidad.

Junto a las descripciones de las entidades y las relaciones propuestas, se adjunta el diagrama relacional en la Figura 4.5 para completar de forma precisa la definición del modelo de datos. El modelo E-R propuesto para las bases de datos de las entidades está formado por un total de ocho entidades conceptuales:

- Entidad Group: Esta entidad hace referencia al conjunto de usuarios de una determinada entidad investigadora que trabajan de forma conjunta sobre una serie de experimentos.
- Entidad User: La entidad User recoge el concepto de personaje que accede a la plataforma, bien sea un usuario de administración o de investigación.
- Entidad Message: La entidad Message hace referencia a los mensajes privados de la entidad de investigación.
- Entidad Experiment: Un experimento biométrico queda representado dentro del modelo relacional por la entidad Experiment.
- Entidad Report: La entidad Report modela los informes de resultados generados tras la ejecución de un experimento.
- Entidad Assembly: La entidad Assembly modela las librerías de código que necesita un proceso privado para realizar sus operaciones

- Entidad Process: La presente entidad define a los procesos creados por los miembros de la entidad de investigación y que por lo tanto, son de carácter privado.
- Entidad Storage: La entidad Storage recoge el concepto de almacén de información biométrica generado tras la ejecución de un experimento. De esta forma, los almacenes privados quedan modelados por la presente entidad.

Una vez presentadas las distintas entidades conceptuales que conforman el modelo de datos, a continuación se describen las relaciones existentes entre ellas. Dentro del modelo conceptual, se han definido cinco interrelaciones:

- Relación Group – tiene – User: La presente relación describe la correspondencia que existe entre los grupos de trabajo y sus miembros. La cardinalidad de la relación es (0,N) ya que un grupo de trabajo puede no poseer ningún miembro. Por su parte, la relación User – pertenece a – Group posee una cardinalidad de (0,N) ya que un usuario de la entidad puede no pertenecer a ningún grupo de trabajo.
- Relación User – publica – Message: Mediante la relación actual se asocian los usuarios de la entidad de investigación y los mensajes privados de la misma. La cardinalidad de la relación es (0,N) puesto que un usuario puede publicar varios mensajes. En el camino inverso, la relación Message – pertenece a – User posee una cardinalidad (1,1) ya que un mensaje solamente pertenece a un usuario.
- Relación Group – tiene – Experiment: La actual relación recoge la asociación entre un grupo de trabajo y sus experimentos. La cardinalidad de la relación es (0,N) puesto que un grupo de trabajo puede no tener experimentos asociados. La relación inversa Experiment – pertenece a – Group posee una cardinalidad (1,1) ya que un experimento pertenece a un solo grupo de trabajo.
- Relación Experiment – genera – Report: La presente relación define la asociación entre los experimentos y los informes de prestaciones. La cardinalidad de la relación es (1,N) ya que como mínimo, tras la ejecución de un experimento se genera un informe de prestaciones. En la relación inversa, la cardinalidad de la relación Report – proviene de – Experiment es (1,1) debido a que un informe solamente proviene de un experimento.
- Relación Assembly – pertenece a – Process: Esta relación asocia los procesos biométricos y las librerías necesarias para ejecutarlos. La cardinalidad de la relación es (0,N) puesto que una librería puede no usarse en ningún proceso. En la relación inversa Process – necesita – Assembly, la cardinalidad de la relación es (1,N) debido a que un proceso necesita como mínimo una librería de código.

4.5. INTERFACE DE DATOS

El interface de datos queda definido por el metamodelo desarrollado (véase el Capítulo 3). La representación detallada de un experimento se denomina BED (Biometric Experiment Definition) y su objetivo es ofrecer la descripción completa de las operaciones y datos involucrados en la ejecución de un experimento. Para la implementación de BED se ha empleado XML por las siguientes razones:

- Es posible extender el metamodelo con nuevas etiquetas. Este hecho permite añadir funcionalidades manteniendo la compatibilidad con versiones anteriores.
- El analizador léxico es un componente estándar y no es necesario crear uno específico para cada versión del metamodelo. Este hecho posibilita el empleo de cualquier analizador disponible, se evitan errores y se acelera el desarrollo.
- Es sencillo entender y procesar la estructura del metamodelo puesto que XML está basado en texto.

La sintaxis de los modelos queda definida a través de un esquema XML, cuya representación se puede consultar en el CD adjunto. A continuación se describen sus principales elementos.

Elemento <BED>

El archivo XML posee un elemento raíz denominado <BED>. Dentro de este elemento raíz aparecen las etiquetas <Validation>, <Execution> y <Definition> para describir diferentes aspectos del modelo.

En primer lugar, el elemento <Validation> ofrece los resultados de las operaciones de validación. El siguiente elemento es la etiqueta <Execution>, cuyo cometido es recoger los resultados de ejecución. Su información se encuentra disponible una vez el experimento ha sido ejecutado. Finalmente, la etiqueta <Definition> describe los elementos que forman parte del modelo. Dentro del presente elemento aparecen las etiquetas <Flows> y <Nodes>. El cometido de estas etiquetas es recoger las propiedades de los flujos y nodos respectivamente.

Elemento <Flow>

El elemento <Flow> describe un flujo de datos. Ofrece una lista de componentes que detallan las propiedades de cada uno de los campos que forman parte de un flujo. Es posible diferenciar dos tipos de campos:

- Value: Son valores o datos por sí mismos que se transportan directamente en un flujo. Un campo de este tipo se etiqueta como un elemento de tipo <Value>.

- Reference: No son valores por sí mismos, sino referencias a archivos donde se almacenan datos que pueden ser utilizados por los nodos. Básicamente, un campo de tipo Reference contiene la URL de un recurso y se etiqueta como un elemento <Reference>.

Elemento <Source>

El nodo Source se representa mediante el elemento <Source>. Esta etiqueta posee un total de dos subelementos. Por un lado, el elemento <SDescriptor> describe el contenido del almacén y por otro, el elemento <Mapping> mapea los campos del almacén en los flujos de salida.

Elemento <Storage>

La etiqueta <Storage> describe las propiedades del nodo Storage. Al igual que el nodo anterior, el elemento XML asociado contiene las etiquetas <SDescriptor> y <Mapping>.

Elemento <Sink>

El nodo Sink se define mediante el elemento <Sink>. A diferencia de los anteriores nodos, su esquema XML asociado solamente posee el elemento <SDescriptor>.

Elemento <Tap>

El nodo Tap se modela mediante el elemento <Tap>. Posee un elemento <Mapping> para definir el contenido del flujo de salida y un elemento <WhereSetup> para configurar cada una de las condiciones del filtro de selección.

Elemento <Join>

El elemento <Join> recoge las propiedades del nodo Join y contiene un total de tres elementos. El elemento <Mapping> define la estructura del flujo de salida y el mapeo de campos desde las entradas. La etiqueta <OnSetup> establece los campos sobre los que se aplica la condición de asociación. Por último, el elemento <WhereSetup> configura el filtro aplicado sobre el flujo de salida.

Elemento <Group>

La etiqueta <Group> presenta los elementos <Mapping> y <BySetup> para describir el comportamiento del nodo Group. De igual forma que en los anteriores nodos, la etiqueta <Mapping> implementa la proyección de los vectores de datos. Mientras, la etiqueta <BySetup> contiene la lista de campos que se emplea como criterio de agrupación.

Elemento <Process>

El nodo Process se implementa mediante el elemento XML <Process>. La etiqueta <Process> contiene un total de tres elementos: <PDescriptor>, que

describe los mecanismos de intercambio de información; <Loading>, que determina los campos de datos que recibe el proceso; y <Mapping> que establece los campos presentes a la salida del nodo.

Elemento <SDescriptor>

El contenido de los almacenes de información se describe mediante el uso de la etiqueta <SDescriptor>. Ésta define el conjunto de campos que forman parte de una fuente de datos biométricos.

Elemento <PDescriptor>

El elemento <PDescriptor> define el interface de datos de un proceso. En él se establecen los parámetros de las entradas y las salidas, así como los de configuración.

Elemento <Mapping>

Todos los nodos (salvo el nodo Sink) poseen un elemento común llamado <Mapping>. Este elemento implementa la primitiva proyección y el mapeo de campos desde las entradas a las salidas. Este elemento permite cambiar el nombre y la descripción de los campos mediante los atributos que contiene.

Elemento <WhereSetup>

El elemento <WhereSetup> implementa la primitiva selección. Está presente en los nodos Tap y Join para definir las condiciones del filtro que se aplica sobre los campos del flujo de entrada. Dentro de la etiqueta <WhereSetup> aparece una lista de elementos <Where>, donde cada uno contiene una de las condiciones del filtro.

Elemento <OnSetup>

El elemento <WhereSetup> implementa la primitiva selección. Se encuentra presente en los nodos Tap y Join para definir las condiciones del filtro de selección de los campos del flujo de entrada. Dentro de la etiqueta <WhereSetup> aparece una lista de elementos <Where>, donde cada uno contiene una de las condiciones del filtro.

Elemento <BySetup>

La primitiva de agrupación se implementa mediante el elemento <BySetup>. Esta etiqueta recoge el conjunto de campos que se emplea como criterio de agrupación. Dentro de ella aparece un elemento <By> por cada uno de los campos que forman parte del conjunto.

4.6. INTERFACE DE COMUNICACIÓN

Debido a que los subsistemas de la plataforma poseen contextos de ejecución

independientes, se ha hecho necesaria la implementación de un sistema de comunicación.

La comunicación entre procesos es una función básica de los sistemas operativos [Silberschatz et al, 2006]. Los procesos se pueden comunicar entre sí mediante espacios compartidos de memoria o a través de las rutinas ofrecidas por el sistema operativo. La comunicación entre procesos se puede clasificar según varias propiedades:

- **Sincronía:** La sincronía hace referencia a la espera de los procesos durante la comunicación. Si la comunicación es síncrona, el proceso emisor permanece bloqueado hasta que recibe la respuesta del proceso receptor.
- **Persistencia:** La persistencia establece la disponibilidad del receptor. Si la comunicación es persistente, el proceso receptor no tiene porqué estar operativo al mismo tiempo que se realiza la comunicación. El mensaje se almacena tanto tiempo como sea necesario hasta ser entregado.
- **Tipo:** Si la comunicación es directa, se ha de explicitar el nombre del proceso destino. Sin embargo, si la comunicación es indirecta, el proceso destino recibe los datos a través de un puerto.
- **Simetría:** La simetría hace referencia a la capacidad de comunicación de los procesos. Si la comunicación es simétrica, todos los procesos pueden enviar y recibir mensajes; en cambio, si la comunicación es asimétrica, un proceso actúa como emisor y el resto como receptores.

La comunicación entre el Front-End y el Back-End se ha implementado mediante una tubería. Ésta no es más que una sección de memoria compartida que varios procesos utilizan para su comunicación [MSDN, 2009c]. Por un lado, se encuentra el proceso que la crea (servidor) y por otro lado, aparecen los procesos que se conectan a ella (clientes). Dentro de las tuberías aparecen las tuberías con nombre [MSDN, 2009d] con el objetivo de ofrecer comunicación entre un proceso servidor y uno o varios procesos clientes. La manera de utilizar la tubería es mediante su nombre, de tal forma que cualquier proceso que conozca el nombre de la tubería puede conectarse a ella. Así mismo, cualquier proceso puede actuar como cliente y como servidor en la comunicación.

Para conseguir la máxima independencia entre los dos subsistemas se propone una comunicación asíncrona, no persistente, indirecta y asimétrica:

- **Interface asimétrico:** El Front-End se encarga de generar notificaciones y el Back-End se encarga de recibirlas, por lo que la comunicación es en un único sentido.
- **Interface asíncrono:** El Front-End no espera respuesta por parte del Back-End cada vez que envía una notificación.

- Comunicación no persistente: Las notificaciones son descartadas tanto si el Back-End no está disponible como si se encuentra en ejecución.
- Comunicación indirecta: Para realizar la comunicación es necesario conocer el nombre de la tubería y no basta con conocer el nombre del proceso asociado al Back-End.

El Back-End se encarga de crear la tubería y por lo tanto, actúa como servidor. Por su parte, el Front-End actúa como cliente y la utiliza cada vez que envía una notificación. A continuación se muestran varios diagramas que explican el proceso de comunicación entre ambos subsistemas. Los elementos implicados en la comunicación de los dos subsistemas se detallan a continuación:

- User: Usuario que interactúa con la plataforma.
- Queue: Tabla que contiene las solicitudes de ejecución.
- ExecutionMngr: Componente del Front-End que recoge las funcionalidades de gestión de experimentos.
- Pipe: Objeto que representa la tubería de comunicación.
- Listener: Subproceso que se mantiene a la espera de notificaciones en la tubería.
- Runner: Subproceso encargado de la ejecución de los experimentos.
- Controller: Subproceso controlador de los recursos asignados a la ejecución.

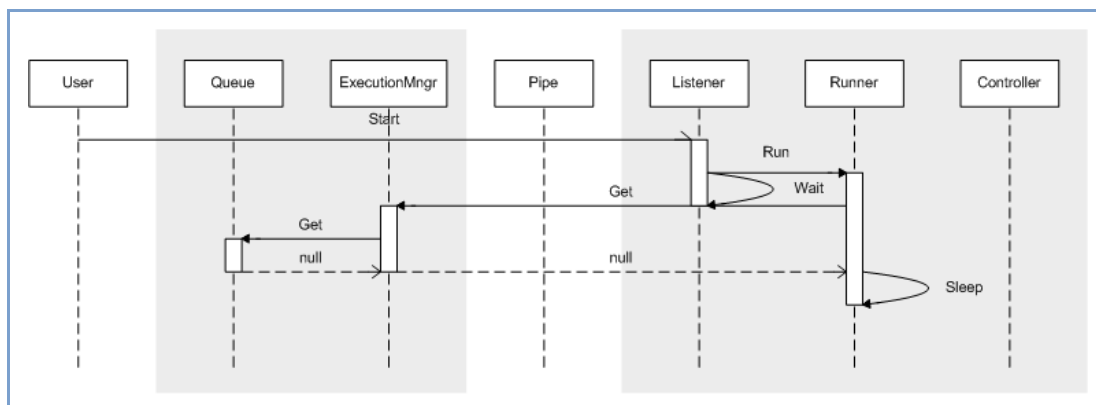


Figura 4.6. Estado Inicial del Interface de Comunicación

Una vez presentados los elementos, a continuación se describen los principales escenarios de comunicación. En la Figura 4.6 se muestra el estado inicial del interface de comunicación. El Back-End se pone en funcionamiento y arranca el subproceso Listener. Éste inicia al subproceso Runner y queda a la espera de una notificación en la tubería. A su vez, Runner comprueba la existencia de solicitudes de ejecución para obtener la información del siguiente experimento y si la cola se encuentra vacía, se duerme.

```

sequenceDiagram
    participant User
    participant Queue
    participant ExecutionMgr
    participant Pipe
    participant Listener
    participant Runner
    participant Controller

    Note over Queue, ExecutionMgr: Phase 1
    User->>Queue: 
    Queue->>ExecutionMgr: Put
    ExecutionMgr->>Pipe: Notify
    Note over Pipe, Listener: 
    Pipe->>Listener: Notify
    Note over Listener, Runner: 
    Listener->>Runner: Get
    Runner->>Runner: Run
    Note over Runner, Controller: 
    Runner->>Controller: Start
    Controller->>Controller: 
    Note over Queue, ExecutionMgr: Phase 2
    Queue->>ExecutionMgr: Request
    ExecutionMgr->>Pipe: Notify
    Note over Pipe, Listener: 
    Pipe->>Listener: Notify
    Note over Listener, Runner: 
    Listener->>Runner: Get
    Runner->>Runner: Wait
    Note over Runner, Controller: 
    Runner->>Controller: Stop
    Controller->>Controller: 
    Note over Queue, ExecutionMgr: Phase 3
    Queue->>ExecutionMgr: Get
    ExecutionMgr->>Pipe: Request
    Note over Pipe, Listener: 
    Pipe->>Listener: Request
    Note over Listener, Runner: 
    Listener->>Runner: Request
    Runner->>Runner: Clear
    Note over Runner, Controller: 
    Runner->>Controller: Clear
    Controller->>Controller: 
  
```

El Back-End puede recibir solicitudes de ejecución de parte del Front-End durante la ejecución de un experimento. En este caso, las notificaciones son desechadas por el subproceso Listener puesto que el Back-End se encuentra ocupado.

La Figura 4.8 muestra la interacción entre los componentes cuando el Back-End no se encuentra disponible. En este caso, el Front-End intenta acceder a una tubería que no existe y se produce una excepción. El control de esta excepción es suficiente para que el Front-End pueda almacenar solicitudes sin necesidad de que el Back-End se encuentre disponible.

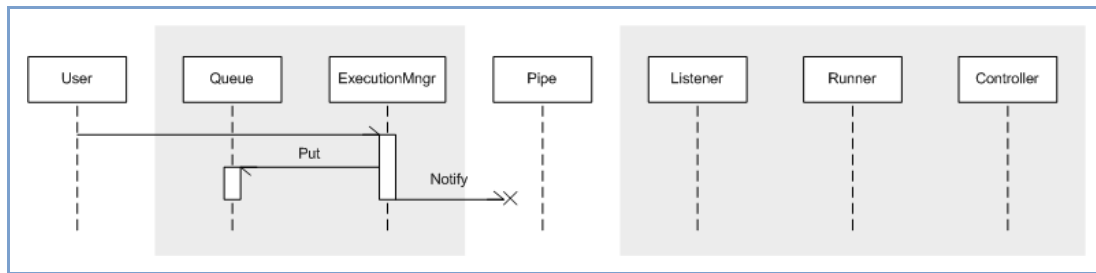


Figura 4.8. Estado de Indisposición del Interface de Comunicación

4.7. DECISIONES TECNOLÓGICAS

En este punto se presentan las tecnologías elegidas para implementar la plataforma. A continuación se exponen las decisiones que han llevado a la elección del entorno de ejecución, del servidor web y del sistema de gestión de las bases de datos.

Microsoft .NET Framework



La plataforma .NET⁶ de Microsoft es un componente software que se puede añadir al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para cubrir las necesidades generales de la programación de aplicaciones haciendo énfasis en la transparencia de redes y la independencia de la plataforma hardware.

.NET se ha tomado como entorno de desarrollo para los subsistemas de la plataforma y como lenguaje de programación se ha elegido C#. Su elección se fundamenta en sus amplias funcionalidades, el número de bibliotecas ya implementadas que ofrece y la capacidad de generar código de forma rápida.

.NET es el encargado de proveer lo que se conoce como código administrado, es decir, un entorno que ofrece servicios automáticos al código que se ejecuta. Los principales servicios que ofrece son un cargador de clases para localizar clases en tiempo de ejecución, un recolector de basura, un motor de seguridad para gestionar la seguridad del código, etc. Soporta más de 20 lenguajes de programación y es posible desarrollar cualquier tipo de aplicación en la plataforma con cualquiera de ellos. Algunos de los lenguajes desarrollados para este entorno de trabajo son C#, Visual Basic, Delphi, C++, J#, Perl, Python, Fortran, Cobol y PowerBuilder.

Aunque C# y .NET son tecnologías propietarias de Microsoft, existe un compilador que provee el Framework de DotGNU⁷ - Mono capaz de generar programas para distintas plataformas como UNIX y Linux por lo que en principio, es posible ofrecer una aplicación multiplataforma.

⁶ <http://www.microsoft.com/.NET/>

⁷ <http://www.gnu.org/projects/dotgnu/>

Internet Information Services



IIS⁸ (Internet Information Services), es una serie de servicios para los sistemas basados en Windows. Convierte a un sistema en un servidor de Internet o Intranet de tal manera que los equipos que tienen este servicio instalado pueden publicar páginas y servicios web tanto de forma local como remota.

IIS ofrece servicios de seguridad y métodos de autenticación que se basan en las últimas tecnologías de cifrado y autenticación mediante certificados de cliente y servidor. Una de las formas que tiene de asegurar los datos es mediante SSL, por lo que se proporciona un método para transferir datos entre cliente y servidor de forma segura.

MySQL



MySQL⁹ es un motor de bases de datos desarrollado por Sun Microsystems que se ofrece bajo licencia GNU GPL para fines no comerciales. Sus principales características son su disponibilidad multiplataforma, la escalabilidad de las operaciones de almacenamiento, y el soporte tanto de transacciones como de claves ajenas. Así mismo, ofrece conectividad segura, replicación de tablas y búsqueda e indexación por campos de texto. MySQL se utiliza principalmente en aplicaciones web puesto que su funcionamiento es muy bueno en entornos de lectura intensiva de datos.

MySQL se ha elegido como sistema gestor de las bases de datos de la plataforma por su escalabilidad, conectividad, versatilidad y disponibilidad como software libre. La única limitación en cuanto al tamaño de una base de datos viene determinada por los recursos del sistema. MySQL permite gestionar bases de datos del orden de seis mil tablas y alrededor de cincuenta millones de registros.

Open Data Base Connector

ODBC¹⁰ (Open Data Base Connector) es un estándar de acceso a Bases de datos desarrollado por Microsoft. Su objetivo es hacer posible el acceso a cualquier dato sea cual sea la aplicación con independencia del sistema de gestión de la base de datos.

ODBC se ha empleado como driver de conexión para aislar el acceso a la información biométrica localizada en orígenes de datos como Microsoft SQL Server y Microsoft Access.

⁸ <http://www.iis.net/>

⁹ <http://www.mysql.com/>

¹⁰ [http://msdn.microsoft.com/en-us/library/ms710252\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms710252(VS.85).aspx)

XML

<?xml?> XML¹¹ es un metalenguaje extensible de etiquetas desarrollado por el W3C (World Wide Web Consortium). No es realmente un lenguaje en particular sino una manera de definir lenguajes para diferentes necesidades. Se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas y no solo para su aplicación en Internet.

XML se ha utilizado en el modelado de los experimentos porque es una tecnología muy sencilla que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

¹¹ <http://www.w3.org/TR/xml-infoset/>

FRONT-END. ANÁLISIS

A lo largo del actual capítulo se ofrece el análisis funcional para el Front-End. El objetivo es la obtención de una especificación detallada a partir de los requisitos de capacidad y de restricción impuestos (véase la sección 4.2). El capítulo comienza con la identificación de los stakeholders y la extracción de los requisitos funcionales del Front-End, continúa con la presentación de los perfiles de usuario y finaliza con la descripción de los casos de uso.

5.1. INTRODUCCIÓN

La Ingeniería de Requisitos se enfoca en la generación de especificaciones para describir de forma clara, consistente y compacta el comportamiento de un sistema [Boehm et al, 2007]. Su principal objetivo es minimizar los problemas relacionados con el desarrollo de sistemas.

Un requisito de software es una necesidad de un usuario para resolver un problema o alcanzar un objetivo. Se puede considerar requisito a aquella condición o capacidad que debe estar presente en un sistema o en algún componente del sistema para satisfacer un contrato, estándar, especificación u otro documento formal [IEEE, 1998].

5.2. IDENTIFICACIÓN DE STAKEHOLDERS

El primer paso del análisis funcional consiste en identificar a los usuarios potenciales de la plataforma y sus necesidades.

Los usuarios a los que se dirige la plataforma son principalmente investigadores del campo de la biometría, cuyas necesidades se centran en la evaluación de sistemas biométricos. No obstante, se puede pensar en cualquier usuario de Internet como usuario potencial de la plataforma.

Así mismo, existen usuarios para dar soporte al Front-End. Aunque no son receptores de los servicios de la plataforma, su interacción con ella es necesaria para su gestión y mantenimiento.

5.3. DEFINICIÓN DEL SISTEMA

El Front-End es el subsistema de la plataforma responsable de la interacción con el usuario. Los requisitos funcionales exigidos por los usuarios se enumeran a continuación:

- La aplicación debe ofrecer servicios con independencia de la ubicación del usuario.
- La aplicación debe permitir a los usuarios la creación y modificación de modelos de experimentos de acuerdo al metamodelo propuesto.
- La aplicación debe ofrecer mecanismos de administración para los experimentos biométricos y sus informes de resultados.
- La aplicación ha de posibilitar la gestión de la información de los usuarios y la comunicación con los investigadores.
- La aplicación debe controlar el acceso concurrente a los modelos con el fin de evitar su alteración accidental o la pérdida de información.
- La aplicación debe ser capaz de presentar los informes de resultados obtenidos tras la ejecución de un experimento.
- La aplicación ha de soportar la modelización de algoritmos para incorporar su descripción a los modelos de los experimentos.
- La aplicación debe permitir el modelado de cualquier tipo de dato biométrico con el fin de agregar su descripción a los modelos.
- La aplicación debe proteger la información individual de los donantes de las bases de datos para asegurar que no es accedida o modificada por usuarios no autorizados.

A partir de estas características se ha desarrollado la definición formal de los requisitos funcionales del Front-End. En el *ANEXO B: ESPECIFICACIÓN DE REQUISITOS* se enumeran los requisitos funcionales forma detallada.

5.4. USUARIOS DEL FRONT-END

A continuación se describen los perfiles y roles de usuarios definidos para el Front-End con el fin de modelar a los stakeholders y sus necesidades.

5.4.1. Perfiles de Usuario

Un perfil de usuario es un modelo que define características básicas de un usuario tales como su tipo, sus permisos y en definitiva, las funcionalidades que un sistema software puede ofrecerle [Amber, 2004]. Para el Front-End se han definido tres perfiles de usuario:

- Perfil de Usuario de Internet: Define a cualquier persona que llega la página de bienvenida de la aplicación mediante un explorador de Internet.
- Perfil de Usuario de Investigación: Modela a los usuarios que utilizan los servicios de la plataforma para evaluar sus sistemas biométricos.
- Perfil de Usuario de Administración: Representa a los usuarios encargados de mantener y gestionar la aplicación para dar servicio a los perfiles anteriores.

5.4.2. Roles de Usuario

Un rol de usuario recoge las funcionalidades que un sistema software ofrece a un usuario bajo ciertas condiciones [Amber, 2004]. Un rol no representa necesariamente a una persona en particular, sino más bien una tarea frente al sistema. A continuación se detallan los diferentes roles de usuario:

- Rol de Usuario de Internet: Describe las funcionalidades para los usuarios que no poseen cuenta en la aplicación o no se han validado en ella. Este rol solamente puede acceder a la página de bienvenida de la aplicación puesto que su proceso de verificación no se ha realizado.
- Rol de Usuario de la Aplicación: Define las operaciones comunes de los usuarios del Front-End. Recoge funcionalidades tales como el control de acceso a la aplicación, la gestión de la información personal y aquellas relacionadas con el sistema de ayuda.
- Rol de Usuario de Grupo de Trabajo: Representa la interacción con la aplicación de los usuarios que pertenecen a uno o más grupos de trabajo dentro de una entidad investigadora. Modela funcionalidades como el control de acceso a los grupos de trabajo y la gestión de los experimentos biométricos.
- Rol de Usuario Propietario: El rol del usuario propietario define la interacción entre el usuario y la aplicación en las funcionalidades de gestión de la entidad investigadora. Describe las funcionalidades para la creación y eliminación de grupos de trabajo y para la gestión de los algoritmos y almacenes de información privados de la entidad.
- Rol de Usuario de Administración: Define la interacción del usuario y la aplicación en las funcionalidades de gestión y mantenimiento de ésta.

5.4.3. Asociación de Perfiles y Roles

Las relaciones entre los perfiles y los roles de usuario se pueden describir mediante un contexto formal [Wolff, 1993]. En el contexto formal establecido, los perfiles y los roles de usuario representan objetos y atributos respectivamente. De esta forma, un perfil de usuario queda definido funcionalmente por todos aquellos roles que puede desarrollar (definición por intensión). La Tabla 5.1 recoge las relaciones entre los perfiles y los roles.

<div>ROL</div> <div>PERFIL</div>	U. de Internet	U. de Aplicación	U. de Grupo de Trabajo	U. Propietario	U. de Administración
U.de Internet	X				
U. de Investigación	X	X	X	X	
U. de Administración	X	X	X	X	X

Tabla 5.1. Contexto Formal para los Perfiles y Roles de Usuario

El Perfil de Usuario de Internet representa a los usuarios no registrados en la aplicación y por lo tanto, solamente pueden realizar las funcionalidades definidas por el Rol de Usuario de Internet.

El Perfil de Usuario de Investigación implementa las funcionalidades definidas para el perfil anterior. Puede desarrollar también las funcionalidades establecidas por los Roles de Usuario de Aplicación, de Grupo de Trabajo y Propietario según sus permisos dentro de la entidad investigadora.

El Perfil de Usuario de Administración implementa todos los roles de usuario.

5.5. CASOS DE USO

La exploración de los requisitos es vital para construir de forma satisfactoria un sistema que cubra las necesidades de los usuarios. Este hecho hace crítico conocer el comportamiento de los usuarios en su interacción con el sistema.

Mediante los casos de uso se describe, desde el punto de vista del usuario, el trabajo del sistema para ofrecer una determinada funcionalidad [Fernández, 2001]. Por un lado, los modelos de casos de uso se centran en la representación de los requisitos que ha de cumplir el sistema frente a la utilización de los usuarios [Amber, 2004]. Por otro, a través de los diagramas de casos de uso se muestra la totalidad de actores y casos de uso del sistema, las interacciones entre ambos y los límites de cada uno de los subsistemas que forman parte del sistema [Kobrun, 2001].

5.5.1. Diagramas de Casos de Uso

En las siguientes páginas se recogen los diagramas de casos de uso para

representar la interacción de los roles de usuario con el Front-End. Las funcionalidades se han dividido en cinco paquetes con el fin de agruparlas según su cometido. La sección 5.5.2 recoge la descripción detallada de cada uno de los casos de uso presentados a continuación.

Gestión de Usuarios

La gestión de usuarios hace referencia a todos aquellos mecanismos encargados de manejar la información de los usuarios de la aplicación. La Figura 5.1 muestra el diagrama de casos de uso para el presente grupo de funcionalidades. Los casos de uso contenidos en este grupo se han identificado mediante el código UM-xxx¹².

Gestión de Entidades

La gestión de entidades recoge a las funcionalidades que se han definido para mantener y gestionar la información de las entidades de investigación. En la Figura 5.2 se muestra el diagrama de casos de uso del presente paquete funcional. Los casos de uso relacionados con la gestión de las entidades de investigación se ha etiquetado mediante el código EM-xxx.

Gestión de Experimentos

El tercer paquete de funcionalidades se corresponde con la gestión de experimentos. Los casos de uso de la gestión de experimentos, cuya identificación utiliza el código XM-xxx, se muestran en la Figura 5.3.

Las funcionalidades de este paquete se reparten entre la aplicación web y el editor de experimentos. Mientras que este último recoge la creación y edición de experimentos, la aplicación web ofrece las restantes funcionalidades.

Gestión de Procesos

La gestión de procesos recoge aquellas funcionalidades relacionadas con la creación, eliminación y modificación de procesos y la gestión de las librerías de código. El diagrama de casos de uso de la Figura 5.4 muestra los casos de uso asociados con este grupo de funcionalidades.

Gestión de Almacenes

El paquete de gestión de los almacenes de información ofrece los mecanismos necesarios para incorporar datos biométricos a las definiciones de los experimentos y gestionar las bases de datos de la plataforma. La Figura 5.5 ofrece los casos de uso asociados con el presente grupo de funcionalidades. La identificación empleada en los casos de uso del presente grupo utiliza el código SM-xxx.

¹² Código numérico de tres cifras.

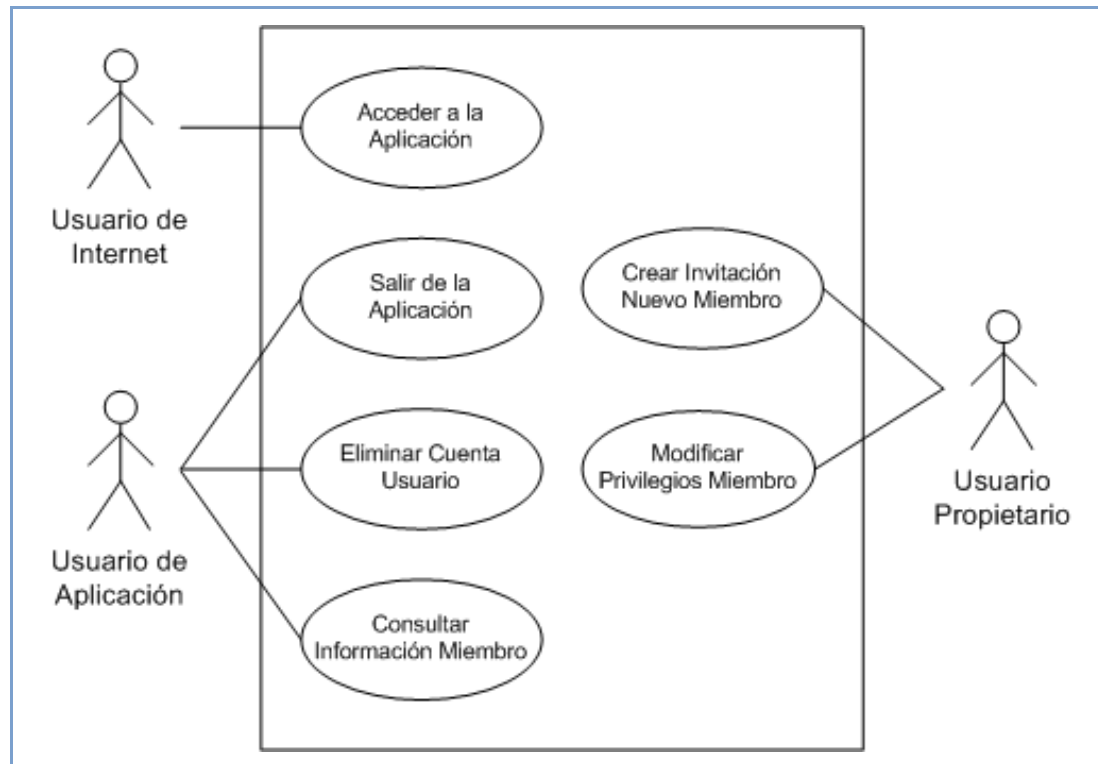


Figura 5.1. Diagrama de Casos de Uso para la Gestión de Usuarios

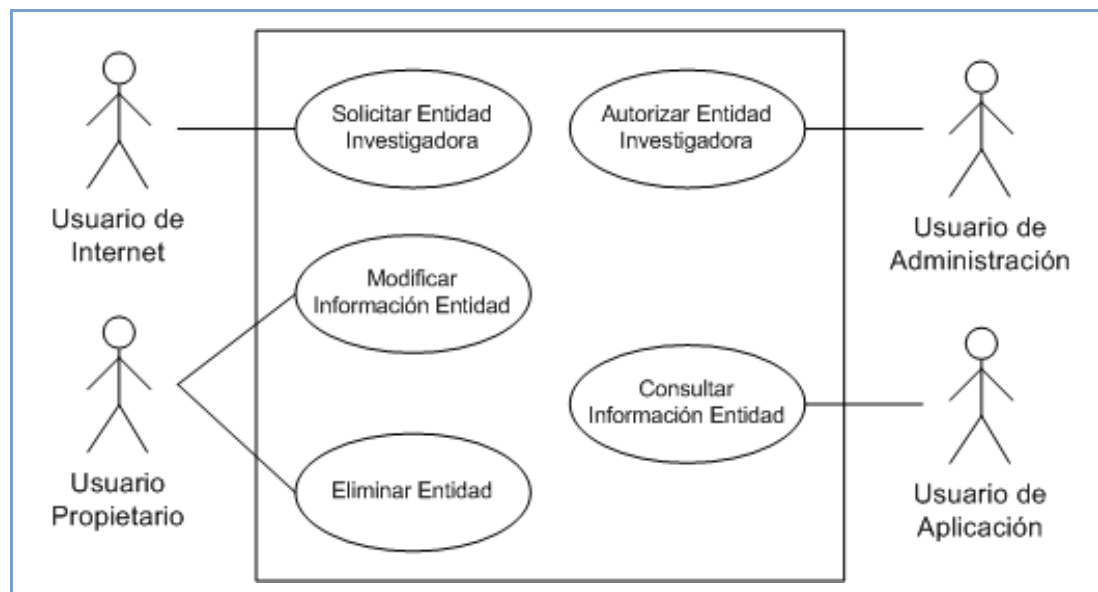


Figura 5.2. Diagrama de Casos de Uso para la Gestión de Entidades

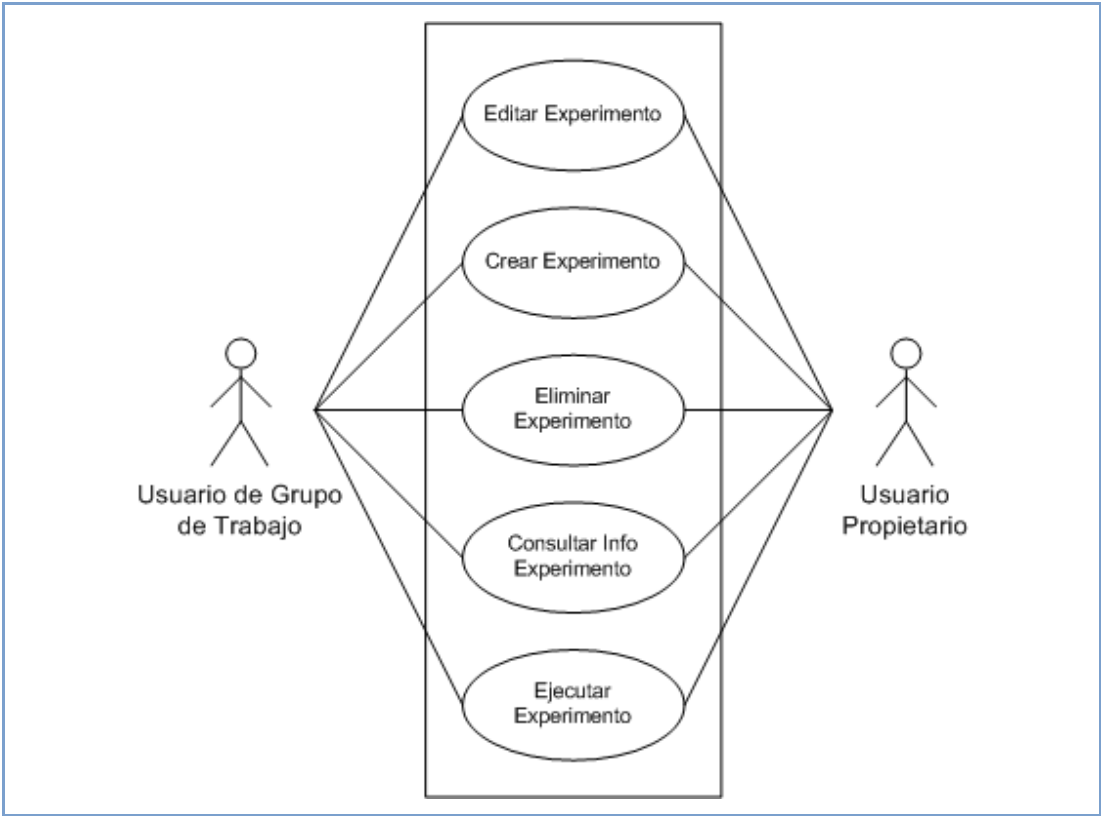


Figura 5.3. Diagrama de Casos de Uso para la Gestión de Experimentos

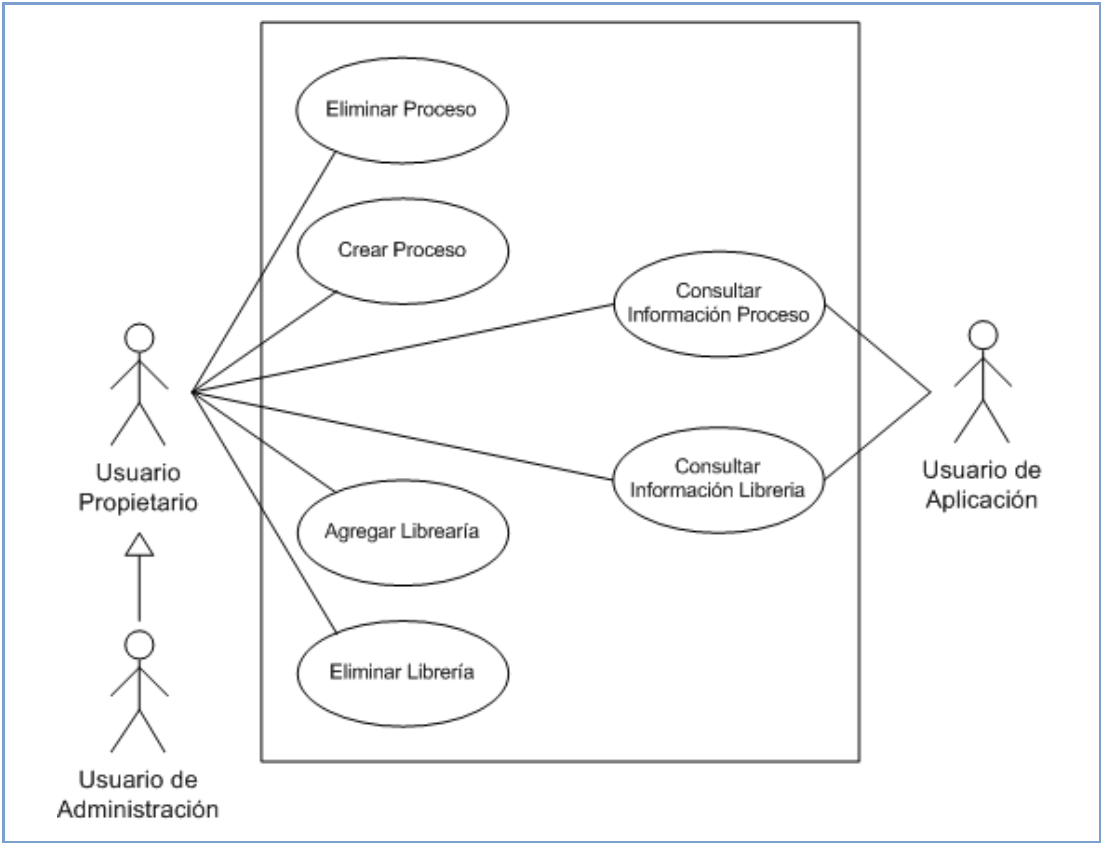


Figura 5.4. Diagrama de Casos de Uso para la Gestión de Procesos

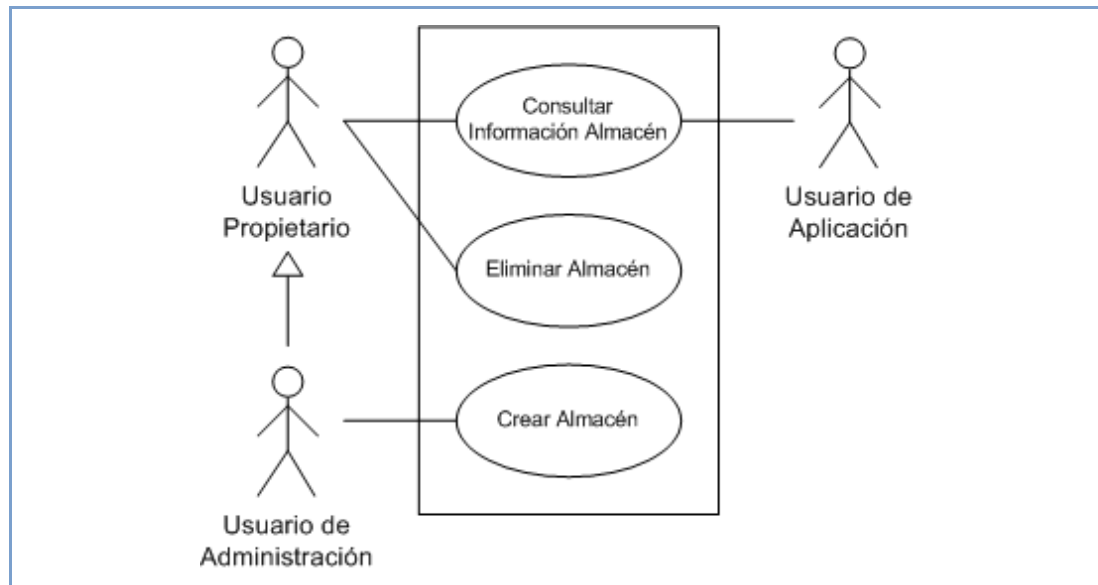


Figura 5.5. Diagrama de Casos de Uso para la Gestión de Almacenes

5.5.2. Descripción de Casos de Uso

A lo largo de las siguientes páginas se muestran las descripciones de los casos de uso introducidos en la sección anterior.

UM-001: Acceder a la Aplicación

DESCRIPCIÓN:

Realiza el proceso de verificación de los usuarios antes de entrar en la aplicación. El proceso de login controla el acceso de los usuarios al sistema, de tal forma que solo aquellos usuarios que validan correctamente su login y su contraseña pueden acceder a la página principal de la aplicación. La función principal de esta operación es evitar el acceso de usuarios no deseados.

ACTORES:

- Usuario de Internet.

PRECONDICIONES:

A fin de permitir el acceso, el actor ha de poseer una cuenta de usuario en la aplicación para que el sistema pueda contrastar los datos facilitados por éste con los existentes en su cuenta.

FLUJO NORMAL:

1. El sistema muestra en la página de bienvenida tres cuadros de texto y un botón para validar los datos.
2. El actor introduce su identificador de usuario, el identificador de su entidad investigadora y su contraseña en los cuadros de texto. A continuación pulsa sobre el botón de validación.
3. El sistema comprueba la validez de los datos. Si los datos son correctos, establece el estado del actor como conectado y muestra el menú principal de la aplicación.

FLUJO ALTERNATIVO:

3. El sistema comprueba la validez de los datos. Si los datos no son correctos, mantiene al actor en la página de bienvenida y muestra un mensaje de error pidiéndole que introduzca los datos de forma correcta.

POSTCONDICIONES:

El actor pasa a estar conectado y se encuentra en la página principal de la aplicación correspondiente a su perfil de usuario.

UM-002: Salir de la Aplicación**DESCRIPCIÓN:**

Permite que el sistema desconecte al usuario. Al cerrar la sesión, el sistema guarda la información que el usuario se encuentra utilizando.

ACTORES:

- Usuario de la Aplicación.

PRECONDICIONES:

El actor ha de estar identificado en el sistema y por lo tanto, tiene que haber realizado previamente el caso de uso UM-001.

FLUJO NORMAL:

1. El sistema muestra un botón para cerrar sesión en todas las páginas que para ser vistas requieren que el usuario se identifique.
2. El actor pulsa sobre el botón para cerrar la sesión.
3. El sistema comprueba si existen operaciones sin terminar por parte del actor. Si no ocurre tal cosa, establece el estado del actor como desconectado y lo lleva a la página de bienvenida de la aplicación.

FLUJO ALTERNATIVO:

3. El sistema comprueba si existen operaciones sin terminar por parte del actor. Si es así, como el proceso de creación de un nuevo almacén, notifica al actor de esta situación y lo mantiene en la página.

POSTCONDICIONES:

Bajo el desarrollo normal de la operación, el actor pasa a estar desconectado y se encuentra en la página de bienvenida de la plataforma.

UM-003: Eliminar la Cuenta de Usuario**DESCRIPCIÓN:**

Permite que el actor elimine su cuenta de usuario. De esta forma, se convierte en un Usuario de Internet y pierde los permisos para acceder a la aplicación.

ACTORES:

- Usuario de la Aplicación.

PRECONDICIONES:

El actor se encuentra en la página de información personal del usuario, dentro de la página principal de la plataforma.

FLUJO NORMAL:

1. El sistema muestra un botón en la barra de herramientas de la página de información personal que permite al actor eliminar su cuenta y darse de baja de la aplicación.
2. El actor pulsa sobre el botón para eliminar su cuenta.
3. El sistema muestra un aviso para que el actor confirme la acción.
4. El actor confirma la acción pulsando el botón para aceptar.
5. El sistema elimina todas las asociaciones del actor con los grupos de trabajo a los que pertenece y su registro. El sistema muestra una notificación al actor de la eliminación satisfactoria de su registro.
6. El actor pulsa sobre el botón aceptar de la notificación.
7. El sistema lleva al actor a la página de bienvenida de la plataforma.

FLUJO ALTERNATIVO:

4. El actor descarta la eliminación pulsando el botón para cancelar.
5. El sistema devuelve al actor a la ficha de usuario.

POSTCONDICIONES:

Si se desarrolla el flujo normal de la operación, la cuenta de usuario del actor se elimina y éste se convierte en un Usuario de Internet.

UM-004: Crear la Invitación de un Nuevo Usuario**DESCRIPCIÓN:**

Permite a un Usuario Propietario invitar a un nuevo miembro a su entidad. La invitación ofrece la posibilidad al nuevo miembro de acceder a un formulario para crear su cuenta de usuario.

ACTORES:

- Usuario Propietario.

PRECONDICIONES:

El actor se encuentra en la página de información de los miembros de la entidad investigadora

FLUJO NORMAL:

1. El sistema muestra un botón en la barra de herramientas que lleva al actor al formulario para la creación de la invitación.
2. El actor pulsa sobre el botón.
3. El sistema muestra un formulario, el cuál consta de un conjunto de cuadros de texto en los que el actor debe introducir el nombre y los apellidos del nuevo usuario junto a la dirección de correo electrónico a la que enviar la invitación. Junto al formulario muestra dos botones, uno para validar y otro para descartar la operación.
4. Una vez introducidos los datos, el actor pulsa sobre el botón para validar los datos.
5. El sistema comprueba la validez de los datos. Si los datos son correctos, genera una nueva invitación y la envía al correo electrónico especificado.

FLUJO ALTERNATIVO 1:

4. El actor pulsa sobre el botón para descartar los datos introducidos en el formulario.
5. El sistema devuelve al actor a la página de información de los usuarios.

FLUJO ALTERNATIVO 2:

5. El sistema comprueba la validez de los datos. Si los datos no son correctos, mantiene al actor en el formulario y le indica los campos erróneos.

POSTCONDICIONES:

Tras el desarrollo de la acción, se crea y se almacena una nueva invitación para el usuario especificado por el actor.

UM-005: Consultar la Información de un Miembro de la Entidad**DESCRIPCIÓN:**

Ofrece al actor la posibilidad de consultar la información de contacto y los permisos de un determinado miembro de su entidad.

ACTORES:

- Usuario de la Aplicación.

PRECONDICIONES:

El actor se encuentra en la página de información de los miembros de la entidad.

FLUJO NORMAL:

1. El sistema ofrece un listado con los identificadores de cada uno de los usuarios que forman parte de la entidad. Para cada miembro muestra su estado y sus privilegios.
2. El actor selecciona un usuario del listado pulsando sobre su identificador.
3. El sistema muestra una nueva página donde ofrece el nombre del usuario, su información de contacto y sus privilegios.

POSTCONDICIONES:

Una vez realizado el desarrollo de la operación, el actor se encuentra en la página de información del usuario seleccionado.

UM-006: Modificar los Privilegios de un Miembro de la Entidad**DESCRIPCIÓN:**

Permite a un Usuario Propietario modificar los privilegios de un determinado miembro de la entidad.

ACTORES:

- Usuario Propietario.

PRECONDICIONES:

El actor debe haber realizado el caso de uso UM-005 con el fin de acceder a la página de información del usuario.

FLUJO NORMAL:

1. Dentro de la página de información del usuario, el sistema muestra una lista que indica los privilegios del usuario en la entidad.
2. El actor selecciona los nuevos privilegios de la lista y pulsa sobre el botón aceptar de la barra de herramientas.
3. El sistema comprueba que la entidad posea propietarios tras el cambio de privilegios. Si es así, actualiza los privilegios del usuario.

FLUJO ALTERNATIVO:

3. El sistema comprueba que la entidad posea propietarios tras el cambio de privilegios. Si no es así, el sistema notifica al actor de la imposibilidad de cambiar los privilegios.

POSTCONDICIONES:

Tras la operación, los privilegios del miembro de la entidad cambian de acuerdo a lo establecido por el actor.

EM-001: Solicitar una Entidad de Investigación**DESCRIPCIÓN:**

Permite a un Usuario de Internet realizar la solicitud de una nueva entidad de investigación. La solicitud es valorada por los administradores de la plataforma, quienes deciden si autorizar o denegar la creación de la entidad.

ACTORES:

- Usuario de Internet.

PRECONDICIONES:

Ninguna.

FLUJO NORMAL:

1. En la página de bienvenida, el sistema muestra un enlace que da acceso al formulario de solicitud.
2. El actor pulsa sobre el enlace.
3. El sistema muestra un formulario que consta de una serie de campos que el actor ha de rellenar y que son utilizados por los administradores de la plataforma para decidir sobre la creación de la entidad. El sistema muestra dos botones debajo del formulario, uno para enviar la solicitud y otro para descartar los datos introducidos.
4. El actor rellena el formulario y al terminar pulsa sobre el botón aceptar.
5. El sistema comprueba la validez de los datos. Si los datos son correctos, almacena la solicitud y envía una copia de la misma a la dirección de correo electrónico especificada en el formulario.

FLUJO ALTERNATIVO:

5. El sistema comprueba la validez de los datos. Si los datos no son correctos, mantiene al actor en la página del formulario y le muestra los datos no válidos.

POSTCONDICIONES:

Tras el desarrollo normal de la acción, el sistema almacena una nueva solicitud de creación pendiente de ser evaluada.

EM-002: Autorizar la Entidad de Investigación**DESCRIPCIÓN:**

Ofrece a un Usuario de Administrador la posibilidad de valorar la propuesta de creación de una determinada entidad de investigación y a continuación, aceptar o denegar su creación.

ACTORES:

- Usuario de Administración.

PRECONDICIONES:

El actor se ha de encontrar en la página de gestión de solicitudes.

FLUJO NORMAL:

1. Dentro la página de gestión de solicitudes, el sistema muestra al actor un listado de todas las peticiones que aún no han sido valoradas.
2. El actor selecciona la solicitud a fin de valorarla.
3. El sistema carga la información de la solicitud y muestra una barra de herramientas que permite realizar diferentes operaciones sobre la propia solicitud.
4. El actor pulsa sobre el botón de la barra de herramientas para aceptar la propuesta.
5. El sistema envía un correo electrónico al usuario solicitante con la información necesaria para acceder al formulario de creación de la entidad.

FLUJO ALTERNATIVO:

4. El actor pulsa sobre el botón de la barra de herramientas para denegar la propuesta.
5. El sistema envía un correo electrónico al usuario solicitante donde le notifica la denegación de la creación de la entidad.

POSTCONDICIONES:

Ninguna.

EM-003: Consultar la Información de la Entidad de Investigación**DESCRIPCIÓN:**

Muestra a un Usuario de la Aplicación la información de su entidad.

ACTORES:

- Usuario de la Aplicación.

PRECONDICIONES:

El actor tiene que haber accedido previamente a la aplicación.

FLUJO NORMAL:

1. Dentro de la página principal de la aplicación, el sistema muestra un enlace que da acceso a la página de información de la entidad.
2. El actor pulsa sobre el enlace.
3. El sistema carga la información de la entidad en una nueva página.

POSTCONDICIONES:

Tras realizarse la operación, el actor se encuentra en la página que muestra la información de la entidad de investigación.

EM-004: Modificar la Información de la Entidad de Investigación**DESCRIPCIÓN:**

El actor puede modificar la información de su entidad investigadora mediante esta operación.

ACTORES:

- Usuario Propietario.

PRECONDICIONES:

El actor tiene que haber realizado previamente el caso de uso EM-003.

FLUJO NORMAL:

1. El sistema muestra en la página un formulario que recoge los datos de la entidad de investigación. En la misma página ofrece una barra de herramientas que permite realizar diferentes operaciones.
2. El actor modifica la información de la entidad y a continuación pulsa sobre el botón para guardar los cambios.
3. El sistema comprueba la validez de los datos. Si los datos son correctos, guarda la información y devuelve al actor a la página de información.

FLUJO ALTERNATIVO:

3. El sistema comprueba la validez de los datos. Si los datos no son correctos, mantiene al actor en el formulario y le indica los errores encontrados.

POSTCONDICIONES:

Tras el desarrollo normal de la operación, la información de la entidad se actualiza de acuerdo a lo establecido por el actor

EM-005: Eliminar la Entidad de Investigación**DESCRIPCIÓN:**

Permite al actor eliminar su entidad.

ACTORES:

- Usuario Propietario.

PRECONDICIONES:

El actor tiene que haber realizado previamente el caso de uso EM-003.

FLUJO NORMAL:

1. El sistema muestra un botón en la barra de herramientas de la página que permite al actor eliminar la entidad de investigación.
2. El actor pulsa sobre el botón.
3. El sistema comprueba que no existe ningún usuario trabajando en la entidad. Si no hay ninguno, el sistema muestra un aviso para que el actor confirme la eliminación.
4. El actor pulsa sobre el botón aceptar para confirmar la operación.
5. El sistema elimina toda la información relacionada con la entidad. A continuación notifica al actor de la finalización de la operación y lo lleva a la página principal de la aplicación.

FLUJO ALTERNATIVO:

3. El sistema comprueba que no existe ningún otro usuario trabajando en la entidad. Si existe algún usuario trabajando en la entidad, el sistema avisa al actor de la imposibilidad de realizar la operación.
4. El actor pulsa el botón aceptar mostrado en la notificación.
5. El sistema lleva de nuevo al actor a la página de información de la entidad.

POSTCONDICIONES:

Tras el desarrollo normal de la operación, se elimina toda la información asociada a la entidad investigadora y los miembros de la entidad se convierten en Usuarios de Internet.

XM-001: Crear un Experimento**DESCRIPCIÓN:**

Permite crear el modelo de un nuevo experimento biométrico.

ACTORES:

- Usuario de Grupo de Trabajo.
- Usuario Propietario.

PRECONDICIONES:

El actor tiene que haber iniciado el editor de experimentos puesto que esta operación no se encuentra disponible en la aplicación web.

FLUJO NORMAL:

1. Tras iniciar sesión, la aplicación ofrece una lista de tareas, entre las que aparece la creación de un nuevo experimento.
2. El actor pulsa sobre el icono para crear un nuevo experimento.
3. La aplicación muestra un formulario para introducir la información del nuevo experimento.
4. El actor introduce los datos del nuevo experimento y pulsa el botón para guardar los datos.
5. La aplicación comprueba la validez de los datos. Si los datos son correctos, crea el nuevo experimento en el espacio de trabajo y carga su definición.

FLUJO ALTERNATIVO:

5. La aplicación comprueba la validez de los datos. Si los datos no son correctos, notifica al actor del error encontrado y lo mantiene en el formulario de creación del experimento.

POSTCONDICIONES:

Tras el desarrollo normal de la operación, el actor dispone de una definición en blanco para crear el modelo de un nuevo experimento.

XM-002: Editar un Experimento**DESCRIPCIÓN:**

Ofrece la posibilidad de modificar el modelo de un experimento.

ACTORES:

- Usuario de Grupo de Trabajo.
- Usuario Propietario.

PRECONDICIONES:

El actor tiene que haber iniciado la aplicación de edición de experimentos puesto que esta operación no se encuentra disponible en la aplicación web.

FLUJO NORMAL:

1. Tras iniciar sesión, la aplicación ofrece una lista de tareas, entre las que aparece la edición de los modelos existentes.
2. El actor pulsa sobre el icono para editar un experimento existente.
3. La aplicación muestra un cuadro de diálogo donde se indican los grupos de trabajo del actor y los experimentos disponibles en cada uno de ellos.
4. El actor selecciona el experimento que desea editar.
5. La aplicación comprueba si el experimento se encuentra libre. Si es así, obtiene el modelo del espacio de trabajo y lo abre para que pueda ser modificado.

FLUJO ALTERNATIVO:

5. La aplicación comprueba si el experimento se encuentra libre. Si no es así, notifica al actor de la imposibilidad de obtener el modelo.

POSTCONDICIONES:

El actor puede modificar el modelo del experimento tras el desarrollo normal de la operación. Por su parte, el experimento queda bloqueado en el espacio de trabajo hasta que sea liberado por el actor.

XM-003: Consultar la Información de un Experimento**DESCRIPCIÓN:**

Permite al actor ver la información de los diferentes experimentos contenidos en un grupo de trabajo.

ACTORES:

- Usuario de Grupo de Trabajo.
- Usuario Propietario.

PRECONDICIONES:

Para realizar la operación, el actor debe tener permisos de acceso al grupo de trabajo y se ha de encontrar en la página de experimentos del grupo.

FLUJO NORMAL:

1. El sistema ofrece una lista donde muestra la información básica de los experimentos disponibles en el grupo de trabajo.
2. El actor navega por la lista de experimentos y pulsa sobre el nombre del experimento
3. El sistema recopila la información del experimento y la muestra en una nueva página.

POSTCONDICIONES:

Tras el desarrollo de la operación, el actor se encuentra en la página de información del experimento.

XM-004: Ejecutar un Experimento**DESCRIPCIÓN:**

Ofrece la posibilidad de enviar un experimento a la cola de ejecución de la plataforma.

ACTORES:

- Usuario de Grupo de Trabajo.
- Usuario Propietario.

PRECONDICIONES:

Para efectuar esta operación, el actor tiene que haber realizado el caso de uso XM-003 y ha de poseer los permisos necesarios dentro del grupo de trabajo. Así mismo, el experimento no se puede encontrar en ejecución ni en uso por ningún otro usuario.

FLUJO NORMAL:

1. El sistema ofrece una barra de herramientas en la página de información del experimento que contiene un botón para enviar el experimento a la cola de ejecución. Este botón solamente está presente si el actor posee los permisos necesarios.
2. El actor pulsa sobre el botón.
3. El sistema comprueba el estado del experimento. Si el experimento se encuentra libre, el sistema procede a ponerlo en la cola de ejecución y a continuación, notifica al actor de la realización de la operación.
4. El actor pulsa sobre el botón mostrado en la notificación.
5. El sistema devuelve al actor a la página de información del experimento.

FLUJO ALTERNATIVO:

3. El sistema comprueba el estado del experimento. Si el experimento no se encuentra libre, notifica al actor de la imposibilidad de enviar el experimento a la cola de ejecución.

POSTCONDICIONES:

Si se produce el desarrollo normal, el sistema pone el experimento seleccionado en la cola de ejecución y éste queda bloqueado.

XM-005: Eliminar un Experimento**DESCRIPCIÓN:**

Permite al actor eliminar un experimento del espacio de trabajo de la entidad de investigación.

ACTORES:

- Usuario de Grupo de Trabajo.
- Usuario Propietario.

PRECONDICIONES:

Para poder efectuar esta operación, el actor tiene que haber realizado el caso de uso XM-003 y ha de poseer los permisos necesarios dentro del grupo de trabajo. Para conseguir el desarrollo normal de la operación, el experimento no se puede encontrar en ejecución ni en uso por ningún usuario.

FLUJO NORMAL:

1. El sistema ofrece una barra de herramientas en la página de información del experimento que contiene un botón para eliminar el experimento. Este botón solamente está presente si el actor posee los permisos necesarios.
2. El actor pulsa sobre el botón.
3. El sistema comprueba el estado del experimento. Si el experimento se encuentra libre, procede a eliminarlo y a continuación, notifica al actor de la eliminación del experimento.
4. El actor pulsa sobre el botón mostrado en la notificación.
5. El sistema devuelve al actor a la página de experimentos del grupo de trabajo.

FLUJO ALTERNATIVO:

3. El sistema comprueba el estado del experimento. Si el experimento no se encuentra libre, notifica al actor de la imposibilidad de eliminar el experimento.

POSTCONDICIONES:

Si se produce el desarrollo normal, el experimento seleccionado es eliminado del espacio de trabajo del grupo.

PM-001: Consultar la Información de una Librería de Código**DESCRIPCIÓN:**

Permite al actor consultar la información de cualquier librería de código almacenada en el espacio de trabajo de la entidad.

ACTORES:

- Usuario de la Aplicación.

PRECONDICIONES:

El actor debe encontrarse en la página de información de las librerías de código dentro de la página principal de la aplicación.

FLUJO NORMAL:

1. El sistema muestra un listado que recoge todas las librerías disponibles.
2. El actor navega por la lista de librerías y hace clic sobre la que desea consultar su información.
3. El sistema abre una nueva página donde carga toda la información relacionada con la librería.

POSTCONDICIONES:

Tras el desarrollo de la operación, al actor se encuentra en la página de propiedades de la librería de código.

PM-002: Agregar una Librería de Código**DESCRIPCIÓN:**

Ofrece al actor la posibilidad de agregar una nueva librería de código al espacio de trabajo de la entidad.

ACTORES:

- Usuario Propietario.

PRECONDICIONES:

El actor debe encontrarse en la página de información de las librerías de código, dentro de la página principal de la aplicación.

FLUJO NORMAL:

1. El sistema ofrece en la página de información una barra de herramientas que permite realizar diferentes operaciones
2. El actor pulsa sobre el botón para agregar una nueva librería.
3. El sistema muestra un formulario que contiene un cuadro de texto para seleccionar el archivo de la librería y una serie de campos adicionales para describir la librería.
4. El actor selecciona el archivo a subir y rellena los campos del formulario.
5. El sistema comprueba que los datos son correctos. Si los datos son correctos, verifica el espacio libre de la entidad de investigación. Si todo es correcto, almacena la librería en el espacio de trabajo y muestra una notificación al actor al finalizar la operación.
6. El actor pulsa sobre el botón aceptar de la notificación.
7. El sistema devuelve al actor a la página de información de librerías.

FLUJO ALTERNATIVO 1:

5. El sistema comprueba que los datos son correctos. Si los datos no son correctos, notifica al actor de los mismos y le mantiene en el formulario.

FLUJO ALTERNATIVO 2:

5. El sistema comprueba el espacio libre de la entidad de investigación. Si no existe suficiente espacio, notifica al actor de la imposibilidad de realizar la operación.

POSTCONDICIONES:

Si se produce el desarrollo normal, la nueva librería se almacena en el espacio de trabajo de la entidad.

PM-003: Eliminar una Librería de Código**DESCRIPCIÓN:**

Permite eliminar una librería de código del espacio de trabajo de la entidad.

ACTORES:

- Usuario Propietario.

PRECONDICIONES:

El actor tiene que haber realizado el caso de uso PM-001 para poder efectuar esta operación.

FLUJO NORMAL:

1. El sistema ofrece en la página de información de la librería una barra de herramientas que contiene un botón para eliminarla. Este botón solamente está presente si el actor posee los permisos necesarios.
2. El actor pulsa sobre el botón.
3. El sistema muestra una notificación sobre la eliminación de la librería de código.
4. El actor pulsa sobre el botón para aceptar la operación.
5. El sistema elimina la librería de código y todos aquellos procesos que la utilizan y notifica al actor tras la finalización de la operación.
6. El actor pulsa en el botón de la notificación.
7. El sistema devuelve al actor a la página de información de las librerías.

POSTCONDICIONES:

La librería seleccionada y todos los procesos que dependen de ella son eliminados del espacio de trabajo tras la ejecución de la operación.

PM-004: Agregar un Proceso**DESCRIPCIÓN:**

El actor puede crear un nuevo proceso a partir de las librerías de código existentes en el espacio de trabajo de su entidad.

ACTORES:

- Usuario Propietario.

PRECONDICIONES:

El actor debe encontrarse en la página de información de los procesos, dentro de la página principal de la aplicación.

FLUJO NORMAL:

1. El sistema ofrece en la página de información una barra de herramientas que permite realizar diferentes operaciones.
2. El actor pulsa sobre el botón para crear un nuevo proceso.
3. El sistema abre una página con un formulario que muestra varios cuadros de texto para seleccionar las librerías que necesita el proceso y una serie de campos adicionales para describirlo.
4. El actor rellena los campos del formulario y pulsa sobre el botón aceptar de la barra de herramientas.
5. El sistema comprueba si los datos son correctos. Si los datos son correctos, almacena la información del nuevo proceso y muestra una notificación al actor al finalizar la operación.
6. El actor pulsa sobre el botón aceptar de la notificación.
7. El sistema devuelve al actor a la página de información de procesos.

FLUJO ALTERNATIVO 1:

5. El sistema comprueba si los datos son correctos. Si los datos no son correctos, notifica al actor de los errores y le mantiene en el formulario.

POSTCONDICIONES:

Si se produce el desarrollo normal se crea un nuevo proceso para la entidad de investigación.

PM-005: Consultar la Información de un Proceso**DESCRIPCIÓN:**

Permite al actor consultar la información de cualquier proceso.

ACTORES:

- Usuario de la Aplicación.

PRECONDICIONES:

El actor debe encontrarse en la página de información de los procesos, dentro de la página principal de la aplicación.

FLUJO NORMAL:

1. El sistema muestra un listado que recoge todos los procesos disponibles, tanto públicos como privados.
2. El actor navega por la lista de procesos y selecciona uno para consultar su información.
3. El sistema abre una nueva página donde carga toda la información relacionada con el proceso.

POSTCONDICIONES:

Tras el desarrollo de la operación, al actor se encuentra en la página de propiedades del proceso.

PM-006: Eliminar un Proceso**DESCRIPCIÓN:**

Permite eliminar un proceso del espacio de trabajo de la entidad.

ACTORES:

- Usuario Propietario.

PRECONDICIONES:

Para poder efectuar esta operación, el actor tiene que haber realizado el caso de uso PM-005.

FLUJO NORMAL:

1. El sistema ofrece una barra de herramientas en la página de información del proceso que contiene un botón para eliminarlo. Este botón solamente está presente si el actor posee los permisos necesarios.
2. El actor pulsa sobre el botón.
3. El sistema muestra una notificación sobre la eliminación del proceso.
4. El actor pulsa sobre el botón para aceptar la eliminación.
5. El sistema elimina el proceso y notifica al actor tras la finalización de la operación.
6. El actor pulsa en el botón de la notificación.
7. El sistema le devuelve a la página de información de los procesos.

POSTCONDICIONES:

El proceso es eliminado tras el desarrollo de la operación.

SM-001: Consultar la Información de un Almacén**DESCRIPCIÓN:**

Permite al actor visualizar la información asociada a un almacén de información biométrica.

ACTORES:

- Usuario de la Aplicación.

PRECONDICIONES:

El actor debe encontrarse en la página de información de los almacenes.

FLUJO NORMAL:

1. El sistema ofrece una lista en la que muestra los nombres de los diferentes almacenes de información disponibles.
2. El actor navega por la lista de almacenes disponibles y pulsa sobre el nombre del almacén cuya información desea visualizar
3. El sistema muestra la información del almacén en una nueva página.

POSTCONDICIONES:

Tras el desarrollo de la operación, el actor se encuentra en la página de información del almacén.

SM-002: Agregar un Almacén de Información**DESCRIPCIÓN:**

Permite agregar un nuevo almacén público de información biométrica a la plataforma.

ACTORES:

- Usuario de Administración.

PRECONDICIONES:

El actor debe encontrarse en la página de información de los almacenes.

FLUJO NORMAL:

1. El sistema muestra un botón en la barra de herramientas que da acceso al formulario de creación de un nuevo almacén.
2. El actor pulsa sobre el botón.
3. El sistema muestra un formulario donde se pueden establecer las propiedades del almacén.
4. El actor selecciona una conexión y una tabla.
5. El sistema recupera la información de la tabla y muestra la lista de campos presentes en la tabla.
6. El actor completa la información de cada uno de los campos y establece los valores de los atributos del almacén. A continuación, pulsa sobre el botón para guardar los datos.
7. El sistema comprueba la validez de los datos. Si los datos son correctos, almacena el descriptor de contenido del almacén y notifica al actor de la finalización de la operación.
8. El actor pulsa sobre el botón mostrado en la notificación y el sistema lo devuelve a la página de información de los almacenes.

FLUJO ALTERNATIVO 1:

5. El sistema recupera la información de la tabla. Si no es posible obtener la información de los campos notifica al actor del error encontrado.

FLUJO ALTERNATIVO 2:

7. El sistema comprueba la validez de los datos. Si los datos no son correctos, notifica al actor de los errores encontrados y lo mantiene en el formulario.

POSTCONDICIONES:

Tras el desarrollo normal de la operación, se crea un nuevo almacén de información biométrica de carácter público.

SM-003: Eliminar un Almacén de Información**DESCRIPCIÓN:**

Permite eliminar un almacén de información de la plataforma, bien sea público o privado.

ACTORES:

- Usuario de Administración.
- Usuario Propietario.

PRECONDICIONES:

El actor tiene que haber realizado previamente el caso de uso SM-001.

FLUJO NORMAL:

1. El sistema ofrece una barra de herramientas en la página de información del almacén que contiene un botón para eliminarlo. Este botón solamente está presente si el actor posee los permisos necesarios.
2. El actor pulsa sobre el botón.
3. El sistema muestra una notificación para confirmar la eliminación.
4. El actor pulsa sobre el botón para aceptarla.
5. El sistema elimina el descriptor de contenido del almacén y su información asociada en caso de ser privado y a continuación, notifica al actor de la realización de la operación.
6. El actor pulsa en el botón de la notificación.
7. El sistema le devuelve a la página de información de los almacenes.

POSTCONDICIONES:

Tras el desarrollo de la operación, el almacén de información deja de estar disponible y no puede ser usado en los modelos de los experimentos.

FRONT-END. DISEÑO

A continuación se ofrece el diseño en detalle del Front-End. Por un lado, el capítulo presenta la definición de su arquitectura y por otro, la especificación detallada de sus componentes.

6.1. INTRODUCCIÓN

Con el diseño se pretende construir un sistema que satisfaga las especificaciones y se ajuste a las limitaciones impuestas [Larman, 2003]. Durante el diseño se presta especial atención en la definición de los objetos software y sus métodos de colaboración con el fin de satisfacer los requisitos propuestos en la etapa de análisis.

6.2. ARQUITECTURA

La definición de la arquitectura del Front-End comprende la partición física del sistema y su organización en subsistemas software.

6.2.1. Arquitectura Física

Tal y como se ha dicho en la sección 4.3, el Front-End responde a una arquitectura cliente-servidor.

6.2.2. Arquitectura Lógica

La arquitectura lógica permite dividir un sistema en subsistemas de diseño con el objetivo de reducir su complejidad y facilitar su mantenimiento. Los patrones arquitectónicos expresan modelos para la organización estructural de los sistemas de software. Proporcionan un conjunto de subsistemas predefinidos, especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos [Buschmann, 1996].

Para la arquitectura lógica del Front-End se propone el modelo de capas ya que se suele aplicar en el diseño de sistemas cuya característica dominante es una mezcla de operaciones de alto y bajo nivel. De esta forma se consigue una estructuración en grupos de subtarefas a diferentes niveles de abstracción

[Buschmann, 1996].

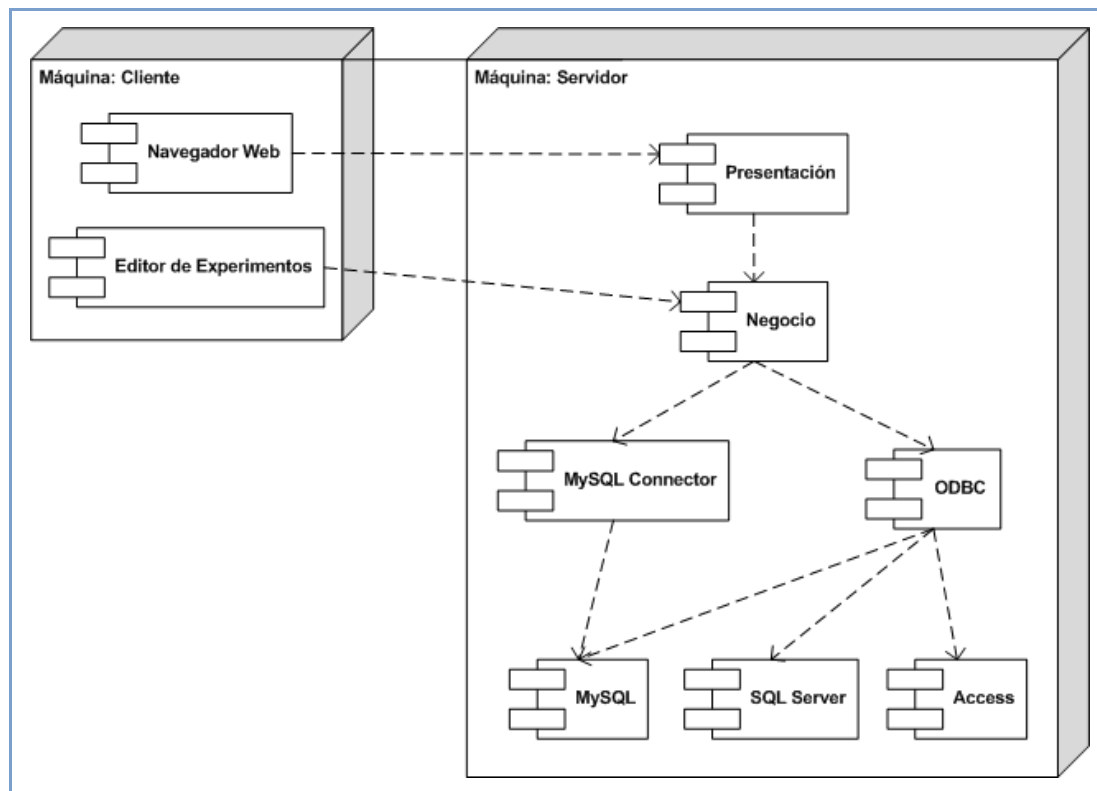


Figura 6.1. Diagrama de Componentes del Front-End

El Front-End, cuyo diagrama de componentes se ofrece en la Figura 6.1, presenta las siguientes capas:

- **Cliente:** Está compuesta por el navegador web y por el editor de experimentos. Para cubrir las funcionalidades relacionadas con la creación y la edición de modelos se propone un subsistema externo a la aplicación web. El objetivo es ofrecer una herramienta cuya funcionalidad no se vea restringida por las propias limitaciones de los navegadores web.
- **Presentación:** Recibe las peticiones de la capa cliente para pasarlas a la capa de negocio y ofrece la vista de la aplicación. La capa de presentación utiliza componentes basados en la tecnología ASP.NET de Microsoft.
- **Negocio:** Procesa las peticiones de la capa de presentación. Es la encargada de ofrecer la lógica de negocio de la aplicación, los servicios de control de concurrencia y el acceso a la capa de persistencia a través de los subsistemas de integración. Esta capa se ha implementado a partir de clases C# estándar.
- **Integración:** Proporciona la comunicación con el subsistema de persistencia. La capa de integración proporciona transacciones, seguridad y escalabilidad en el acceso a los datos. Esta capa está compuesta por MySQL Connector (acceso a los datos de la aplicación) y ODBC (acceso

a las bases de datos biométricos).

- Persistencia: Tiene como finalidad albergar los datos de la aplicación.

6.3. CAPA DE NEGOCIO

La capa de negocio alberga los procesos que ejecutan las operaciones necesarias para ofrecer las funcionalidades de Front-End (véase la sección 5). Se comunica con la capa de presentación para recibir las solicitudes del usuario y mostrar los resultados e invoca a la capa de persistencia para almacenar y recuperar datos de ella.

La capa de negocio del Front-End está compuesta por nueve paquetes, cuyas dependencias se recogen en la Figura 6.2. De los nueve paquetes propuestos, tres ofrecen soporte (SetupMngmnt, Utils y DbMngmnt), cinco recogen las funcionalidades propuestas (UserMngmnt, EntityMngmnt, StorageMngmnt, ProcessMngmnt y ExperimentMngmnt) y uno modela los elementos de los modelos (Elements). En las siguientes páginas se describe cada uno de los paquetes de la lógica de negocio.

Paquete SetupMngmnt

El paquete SetupMngmnt recoge las operaciones relacionadas con la gestión de los parámetros de configuración de la plataforma. El diseño del presente paquete se basa en el patrón Value Object. Un Value Object no es más que un objeto que empaqueta datos y que permite, de forma compacta y organizada, la transferencia de información entre capas.

Paquete Utils

El paquete Utils engloba un conjunto de clases que ofrecen operaciones auxiliares. Este paquete brinda mecanismos para validar y cifrar datos, generar secuencias de caracteres aleatorias y gestionar el envío de correos electrónicos.

Paquete DbMngmnt

El paquete DbMngmnt proporciona un interface unificado para el acceso a los servicios del componente MySQL Connector. En este caso, se ha propuesto un diseño basado en el patrón Facade. La principal motivación para utilizar el paquete DbMngmnt es reducir la complejidad de la capa de negocio y su dependencia con respecto al componente MySQL Connector.

Paquete UserMngmnt

El paquete UserMngmnt recoge a todas aquellas clases que ofrecen la lógica de negocio para gestionar la información de los usuarios, de los grupos de trabajo y de las invitaciones. El patrón de diseño de los elementos de este paquete se sustenta de nuevo en Value Object.

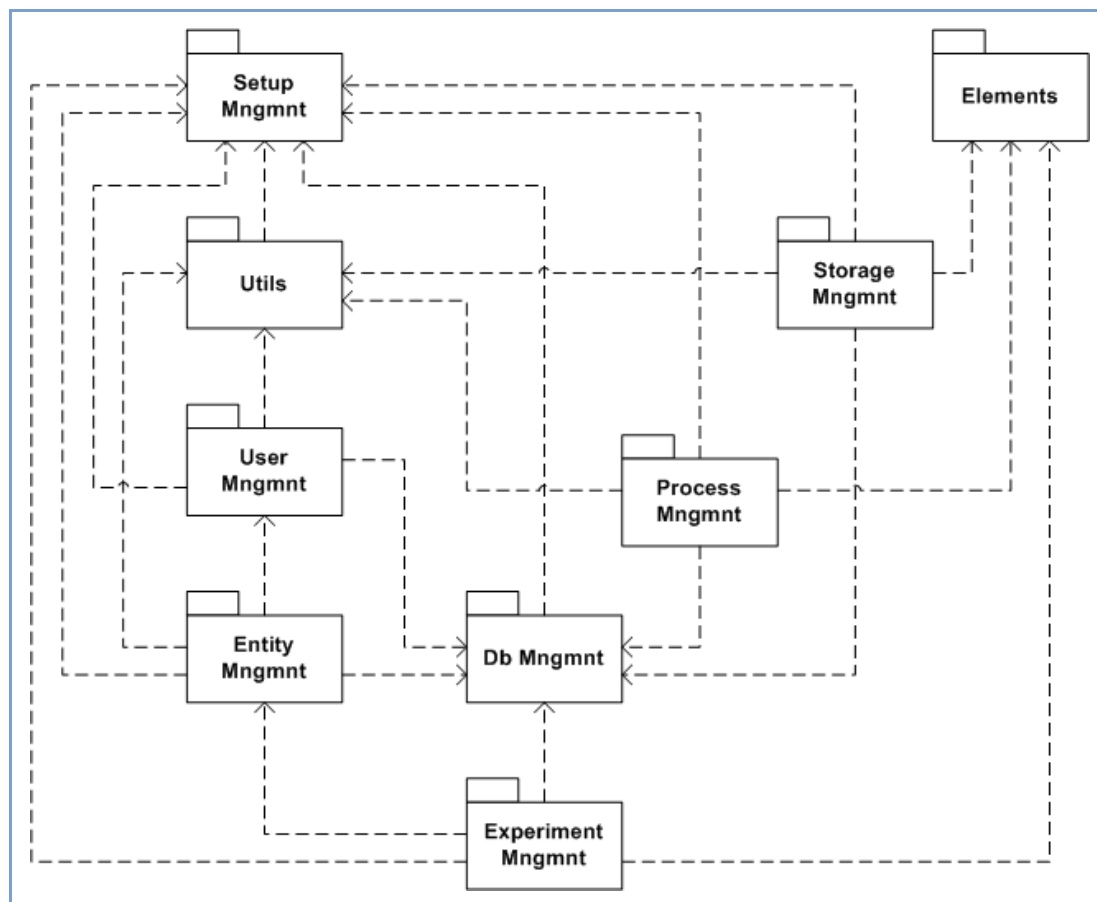


Figura 6.2. Paquetes de la Capa de Negocio

Paquete EntityMngmnt

El paquete EntityMngmnt proporciona la lógica de negocio necesaria para gestionar la información de las entidades de investigación y las solicitudes de creación. Al igual que en el paquete anterior, su diseño parte del patrón Value Object.

Paquete StorageMngmnt

El paquete StorageMngmnt proporciona las clases que implementan la lógica de negocio para la gestión de los almacenes de información biométrica. El diseño del presente paquete se basa en la utilización de diferentes patrones. Por un lado, la transferencia de datos hacia las capas de presentación y de persistencia se basa en el patrón Value Object. Por su parte, la creación de los descriptores de contenido toma como punto de partida el patrón Abstract Factory. La creación de éstos se delega en clases específicas en función del tipo de motor de la base de datos.

Paquete ProcessMngmnt

El paquete ProcessMngmnt contiene la lógica de negocio para gestionar los procesos. Ofrece funciones para crear algoritmos a partir de librerías de código y describir sus interfaces de datos. El diseño del paquete se basa en el patrón

Value Object.

Paquete ExperimentMngmnt

El último paquete ofrece las operaciones necesarias para gestionar la información de los experimentos. Así mismo, alberga los mecanismos de gestión de la cola de ejecución y de acceso al interface de comunicación con el Back-End.

Paquete Elements

El presente paquete alberga la lógica de negocio para los modelos de los experimentos y es compartido por los dos subsistemas de la plataforma.

6.4. CAPA DE PRESENTACIÓN

En la actualidad existe un gran interés en la construcción de sitios web potentes con el objetivo de proporcionar un valor real y una experiencia positiva al usuario. Cuando los visitantes evalúan positivamente su contenido, facilidad de uso, rendimiento, fiabilidad y satisfacción general, se habla de un sitio web centrado en el usuario [Van Duyne et al, 2002].

6.4.1. Mapa Conceptual

El mapa conceptual de la aplicación es un prototipo esencial que recoge las características generales y la estructura básica del interface de usuario. El interface de usuario se ha dividido en tres niveles funcionales jerárquicos: la página de bienvenida, la página principal y las páginas privadas para los diferentes perfiles de usuario.

6.4.1.1. Nivel 1. Página de Bienvenida

El primer nivel del interface es accesible para todos los usuarios puesto que es suficiente con introducir la dirección de la aplicación en cualquier explorador de Internet para llegar a dicho nivel. La función principal de este nivel es controlar el acceso de los usuarios a la aplicación y por ello, recoge los casos de uso UM-001 y EM-001. El nivel superior del interface de usuario corresponde a la página de bienvenida y está representado en la Figura 6.3.

6.4.1.2. Nivel 2. Página Principal

El segundo nivel del interface corresponde a la página principal de la plataforma, cuyo acceso queda reservado a los usuarios de la aplicación. Este nivel se encarga de ofrecer acceso a las funciones de gestión de la información personal, a los menús privados de la plataforma y al sistema de ayuda. La estructura del segundo nivel del interface de usuario viene representada en la Figura 6.4. En el presente nivel se desarrollan los casos de uso UM-002 y UM-003.

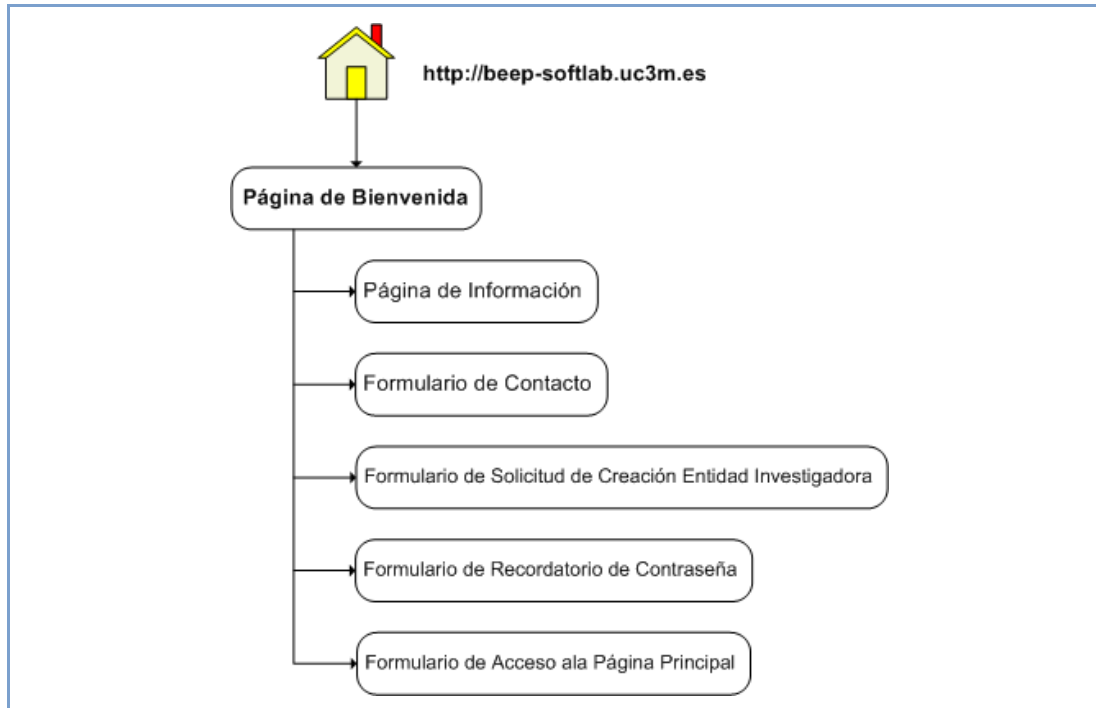


Figura 6.3. Nivel Superior del Interface Gráfico

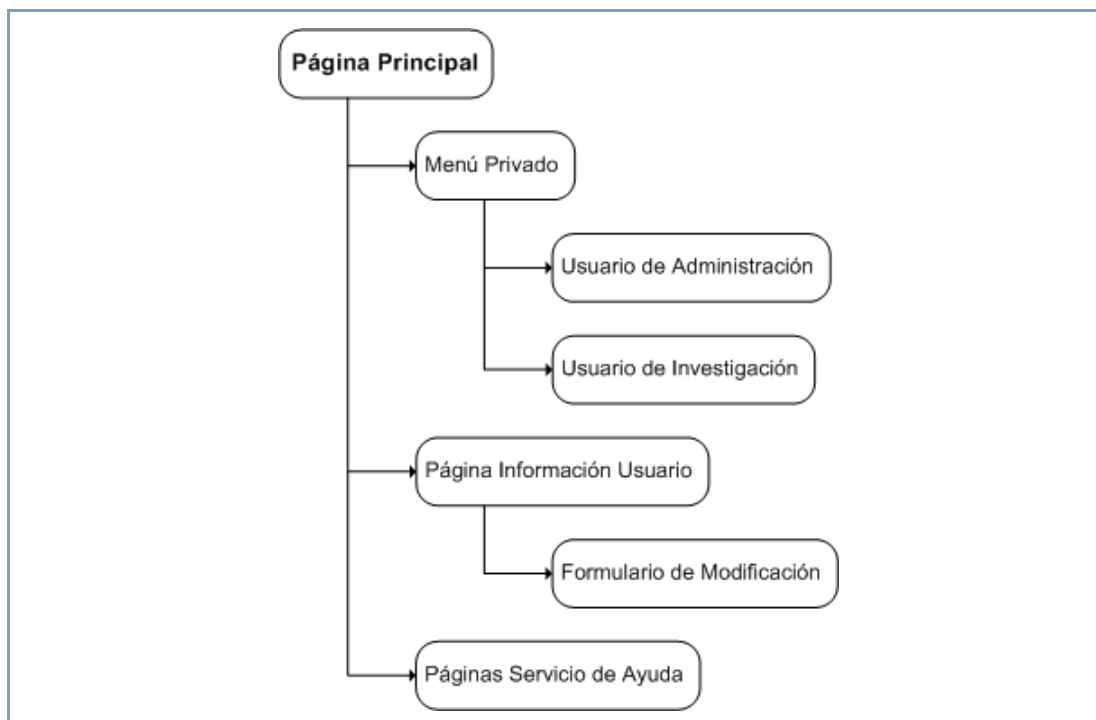


Figura 6.4. Segundo Nivel del Interface Gráfico

6.4.1.3. Nivel 3. Herramientas

El nivel inferior corresponde a las diferentes páginas de herramientas del interface de usuario. El presente nivel es diferente para los perfiles de los usuarios de investigación y administración. Mientras que los niveles superiores se encargan principalmente del control de acceso a la aplicación y de la gestión

la información individual de los usuarios, el nivel inferior se encarga de ofrecer las funcionalidades específicas de administración e investigación a los usuarios.

Herramientas de los Usuarios de Administración

El menú privado para el Perfil de Usuario de Administración ofrece las funciones necesarias para el mantenimiento de la plataforma: gestión de usuarios, de grupos de investigación y de datos biométricos. La Figura 6.5 recoge la estructura del nivel inferior para el Perfil de Usuario de Administración. En el presente nivel funcional aparecen los casos de uso EM-002, SM-001, SM-002, SM-003, PM-001, PM-002, PM-003, PM-004, PM-005 Y PM-006.

Herramientas de los Usuarios de Investigación

El nivel inferior del interface de usuario se completa con las herramientas para el Perfil de Usuario de Investigación. En la Figura 6.5 se representa la estructura completa del nivel; sin embargo, las funcionalidades que ofrece dependen de los roles que asume el usuario.

Funcionalmente, la aplicación de edición de experimentos se encuentra dentro del presente nivel. En este nivel funcional aparecen los casos de uso UM-002, UM-003, UM-004, UM-005, UM-006, EM-003, EM-004, EM-005, XM-001, XM-002, XM-003, XM-004, XM-005, SM-001, SM-003, PM-001, PM-002, PM-003, PM-004, PM-005 Y PM-006.

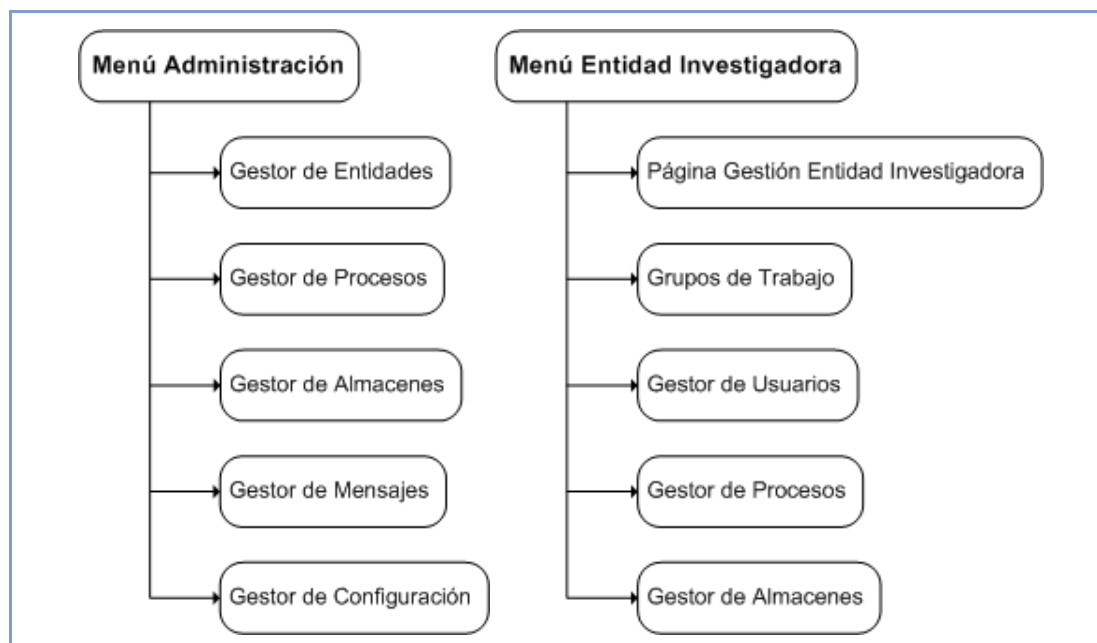


Figura 6.5. Nivel Inferior del Interface Gráfico

6.4.2. Diseño Gráfico

La facilidad de aprendizaje y utilización de un sistema de información es lo que

se denomina usabilidad [Nielsen, 2000]. Conforme aumenta el tamaño y la complejidad de las aplicaciones web se afrontan mayores retos en la creación de sitios funcionales y la usabilidad se convierte en una característica clave. En la actualidad es necesario conocer los principios básicos de la arquitectura de la información, la usabilidad y la ingeniería de la interfaz de usuario [Morville, 2006]. A continuación se ofrecen los principales requisitos de usabilidad durante el ciclo de vida de un sitio web:

- **Eficacia:** El sitio debe ayudar al usuario a conseguir su meta de manera efectiva y eficaz. Los usuarios normalmente no se preocupan por los gráficos de alta calidad y solamente desean alcanzar su objetivo.
- **Control:** Una aplicación web es controlable por el usuario si éste puede influir en la velocidad y la dirección de la aplicación hasta alcanzar su objetivo.
- **Coherencia:** Las convenciones establecidas en el sitio web no deben ser muy diferentes de las establecidas en la en la mayoría de los demás sitios web. Se deben utilizar términos conocidos por los usuarios y evitar nuevas acepciones que puedan confundir al visitante.
- **Personalización:** Un sitio web se entiende como personalizado cuando se ajusta a la medida de las necesidades y características culturales de las personas que lo utilizan.
- **Explicación:** Una aplicación web es explicativa si ayuda al usuario a aprender y entender el sistema. Esto es especialmente importante en la web, donde la mayoría de los usuarios son novatos respecto al sitio que se pretende desarrollar.

La capa de presentación alberga todas las páginas, tanto dinámicas como estáticas, de la aplicación. El diseño gráfico de las páginas toma como punto de partida los principios de usabilidad establecidos. La vista de las páginas se ha diseñado manteniendo un mismo criterio estructural, cuyo objetivo es ofrecer un interface claro y sencillo de utilizar.

Dentro del nivel de presentación se distinguen dos tipos de páginas. Las páginas públicas, cuyo acceso no está restringido, son aquellas del primer nivel funcional. Su estructura se muestra en la Figura 6.6 y está formada por los siguientes elementos:

- **Cabecera:** Muestra el logotipo de la aplicación.
- **Contenido:** Ofrece la información específica de cada página del interface.
- **Pie:** Muestra los enlaces a las páginas de información y de contacto.

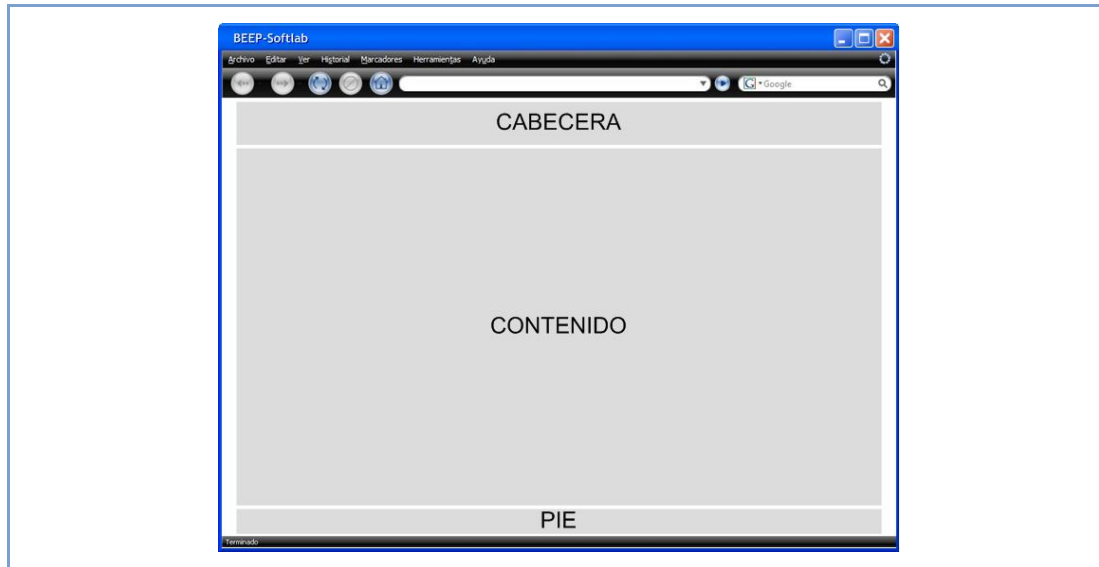


Figura 6.6. Diseño Gráfico de las Páginas Públicas

Las páginas privadas corresponden a los niveles funcionales dos y tres. Su estructura se ha dividido en los siguientes elementos, tal y como muestra la Figura 6.7:

- Cabecera: Al igual que en las páginas públicas, ofrece el logotipo de la aplicación.
- Opciones: Esta zona de la página recoge una barra de botones que da acceso al servicio de ayuda y a la página de información del usuario. Así mismo, ofrece el botón para cerrar sesión.
- Menú: Alberga los enlaces que llevan a las diferentes páginas de la aplicación.

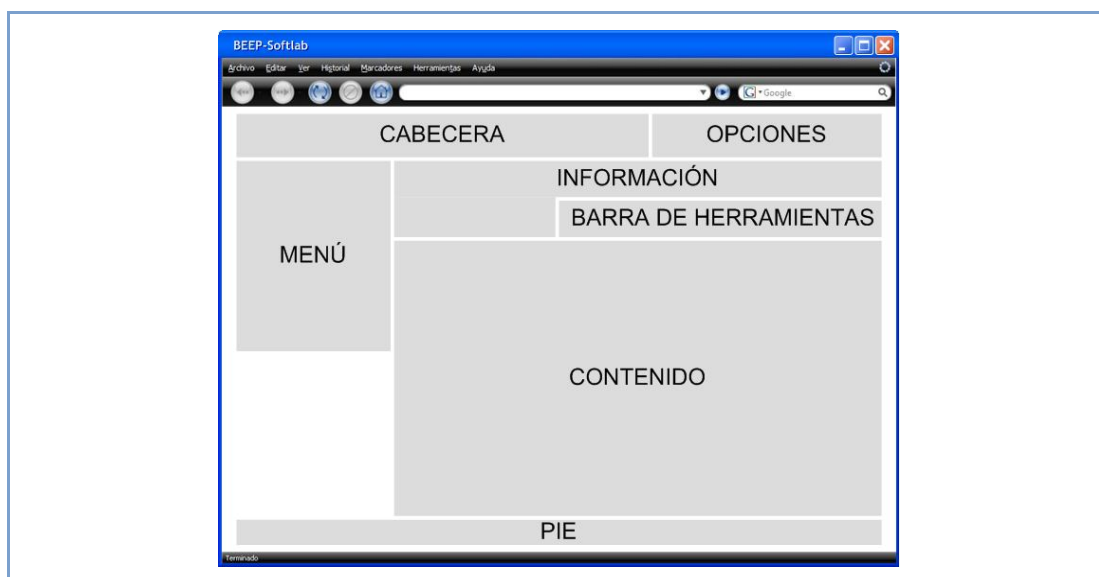


Figura 6.7. Diseño Gráfico de las Páginas Privadas

- Información: Muestra una descripción acerca de las funcionalidades de la página y contiene una miga de pan
- Barra de herramientas: Contiene controles propios para cada una de las páginas.
- Contenido: Ofrece la información específica de cada página del interface.
- Pie: Muestra los enlaces a las páginas de información y de contacto, de la misma forma que en las páginas públicas.

6.5. EDITOR DE EXPERIMENTOS

El editor de experimentos es el componente software encargado de suministrar las funcionalidades de creación y edición de modelos para los experimentos. Su diseño parte de una arquitectura MVC (Model View Controller) [Buschmann, 1996], cuyo objetivo es separar los datos, la interfaz de usuario y la lógica de control de la aplicación. Los componentes del patrón MVC son:

- Modelo: Representación específica de los datos de la aplicación.
- Vista: Interface gráfico encargado de la interacción con el usuario y de la transformación del modelo para que sea visualizado.
- Controlador: Lógica responsable del procesado y del comportamiento según las peticiones del usuario. Acomete la construcción del modelos y lo entrega a la vista para su visualización.

Las principales motivaciones para la utilización del patrón MVC son su mínimo grado acoplamiento entre los componentes y su alto nivel de cohesión. Con su aplicación se consiguen componentes altamente especializados y poco dependientes. La separación que ofrece hace que las modificaciones en cualquier componente conlleven un mínimo impacto sobre los restantes.

Vista

La vista de la aplicación se corresponde directamente con el paquete GUI. Las clases que ofrece son las responsables de la creación del interface gráfico y de la interacción con el usuario.

Controlador

La lógica de control de la aplicación recae sobre los paquetes Elements y Commands.

El paquete Elements contiene la lógica de control necesaria para crear y editar los modelos de los experimentos. La lógica de negocio de éste es mucho más amplia que la del paquete Elements del Front-End ya que es necesario gestionar los elementos visuales y se ha de proveer mecanismos para crear y

validar modelos. El paquete se ha desarrollado haciendo uso de los siguientes patrones de diseño:

- **Abstract Factory:** Proporciona un interface para crear familias de objetos relacionadas y que dependen entre sí. La lectura de un modelo desde el archivo XML se basa en este patrón ya que la creación de los objetos se delega en las clases específicas en función de los nombres de las etiquetas XML.
- **Prototype:** Especifica los tipos de objetos a crear por medio de una instancia prototípica y crea nuevos objetos mediante copias del prototipo. La creación de nuevos nodos toma como base este patrón de diseño.
- **Iterator:** El presente patrón se ha tomado como punto de partida en el mecanismo de validación del modelo ya que ofrece un método para acceder secuencialmente a los elementos de un objeto.

Un efecto común que se produce al dividir un sistema en una colección de clases cooperantes es la necesidad de mantener una consistencia entre objetos relacionados. El paquete Elements define un controlador de eventos encargado de registrar y notificar los cambios de la aplicación. El controlador de eventos, cuya implementación toma el patrón Observer, se emplea para cambiar la vista de los elementos de la aplicación en función de las acciones del usuario.

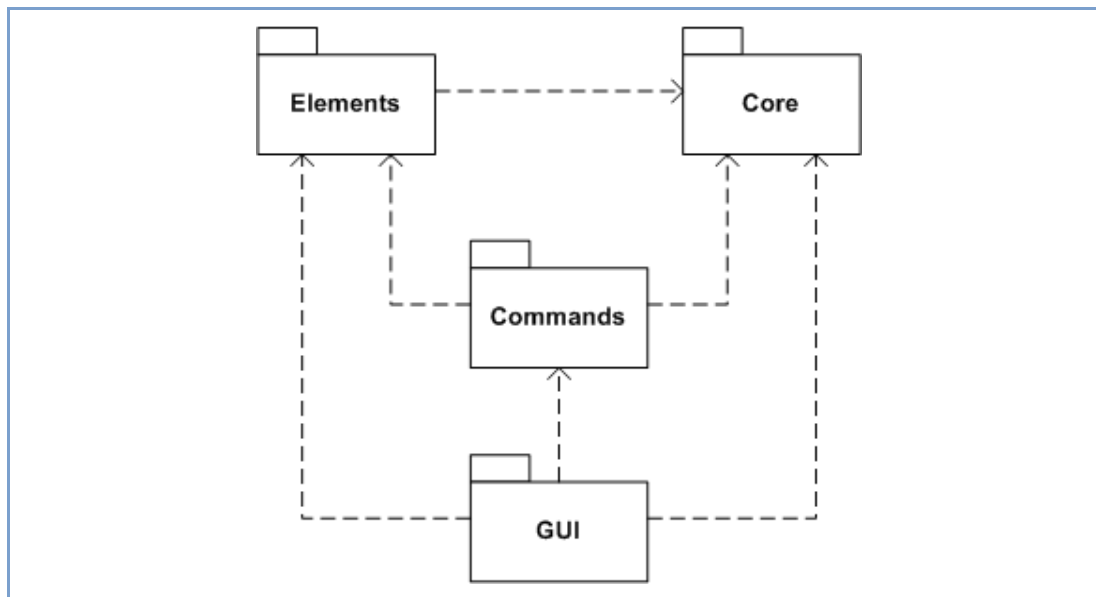


Figura 6.8. Paquetes del Editor de Experimentos

El paquete Commands contiene las clases que implementan las operaciones definidas para el editor de experimentos. Este paquete se ha construido según el patrón de diseño Command con el objetivo de encapsular las peticiones en objetos. De esta forma es posible registrar las operaciones realizadas y se puede ofrecer mecanismos para deshacerlas y rehacerlas.

Modelo

El modelo de la aplicación se corresponde con el paquete Core. Este paquete define las clases que mantienen los datos y el estado de la aplicación. El paquete Core hace uso del patrón Value Object para obtener y establecer los parámetros de configuración.

GESTIÓN DEL PROYECTO

En las siguientes páginas se detalla la planificación y el presupuesto para el proyecto. El capítulo presenta la estimación de los recursos económicos y humanos necesarios y a continuación muestra el plan de utilización de los mismos a lo largo de las diferentes etapas del proyecto.

7.1. INTRODUCCIÓN

La planificación surge de la necesidad de estimar los costes económicos y temporales que requiere el desarrollo de un proyecto así como del control de los factores que pueden alterar su evolución [Humphrey, 1995]. En base al conjunto de funcionalidades propuestas, se hace imprescindible definir un calendario de trabajo con el fin de ordenar la ejecución de cada una de las tareas necesarias para su implementación.

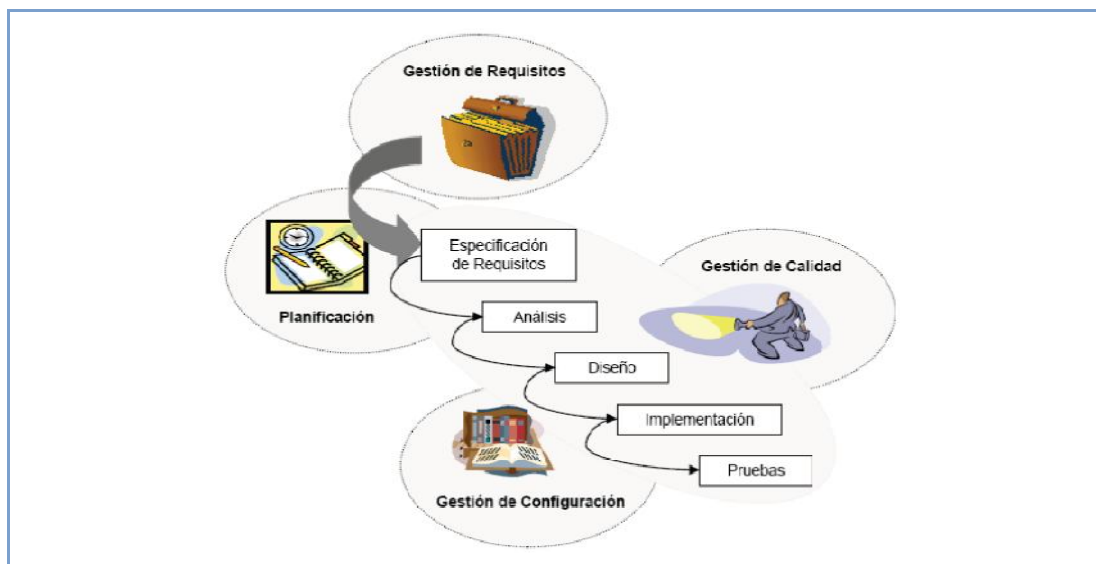


Figura 7.1. Proceso de Desarrollo Software

El proceso software es la descripción de las etapas que se siguen durante la ejecución de un proyecto. Su correcta definición permite asegurar una adecuada asignación de recursos y conocer el estado del proyecto en cada momento. Establece el marco de trabajo, tanto técnico como de gestión, en la

aplicación de los métodos, las herramientas y las personas a las tareas de desarrollo de software [Cuevas, 2003].

Una metodología de trabajo definida permite que cada nuevo proyecto sea construido en base a la propia experiencia y a la de los predecesores. Su utilización permite identificar las causas de los problemas para corregirlos. En la Figura 7.1 se muestra el proceso donde se unen las actividades de desarrollo y gestión. Estas últimas se orientan a controlar el desarrollo del proceso software y a corregir las desviaciones con respecto a los parámetros de calidad establecidos. Los procesos de gestión se ejecutan de manera paralela a las operaciones de desarrollo.

7.2. METODOLOGÍA

El ciclo de vida es el periodo que comienza cuando se concibe un producto software y termina cuando dicho producto deja de estar disponible. Se divide normalmente en fases que estructuran y organizan las etapas de concepción, desarrollo y mantenimiento del sistema software.

Debido a la naturaleza del proyecto se ha utilizado el ciclo de vida en espiral. En este modelo el proyecto se ataca en una serie de ciclos de vida cortos, cada uno de los cuales finaliza con una revisión de software ejecutable.

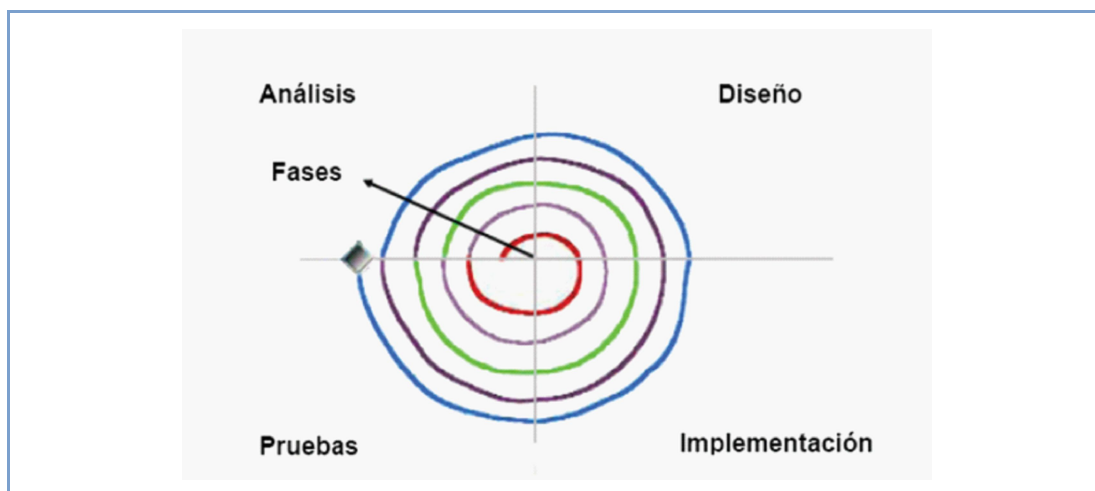


Figura 7.2. Ciclo de Vida en Espiral

En la Figura 7.2 se muestra un ejemplo de desarrollo completado en cinco iteraciones del ciclo. En cada una de éstas se desarrollan todas las fases del ciclo y se incorporan las experiencias de las entregas anteriores, con sucesivas etapas de especificación, construcción, explotación y revisión para obtener un refinamiento del producto de cara al siguiente ciclo.

Este ciclo de vida es especialmente recomendable cuando se planea hacer múltiples entregas del software. Se emplea en proyectos de gran tamaño o larga duración, en aquellos que necesitan tecnologías muy avanzadas o

cuando se requiere la experiencia del usuario para refinar el diseño. Las principales motivaciones que han llevado a su elección son las siguientes:

- Complejidad del sistema: Debido a la complejidad del proyecto y a los recursos disponibles se ha abordado el desarrollo de cada uno de los componentes de la plataforma por separado en varios ciclos. Esto ha permitido corregir anomalías y desviaciones.
- Riesgo tecnológico: El ciclo de vida adoptado permite asumir ciertos riesgos derivados del desconocimiento tecnológico.
- Dinamismo y adaptabilidad: El ciclo de vida acerca el sistema a la solución de forma progresiva y permite la especialización del equipo de desarrollo en función del componente a implementar.

7.3. ESTIMACIÓN DE RECURSOS TEMPORALES

La fecha de inicio del proyecto se establece el día 25 de Septiembre de 2008 y la fecha de finalización en base a la actual planificación se emplaza el día 3 de Julio de 2009. Entre estas dos fechas se sitúa el desarrollo del proyecto con un total de 817 horas. El reparto de horas para cada una de las tareas del proyecto se presenta en la Tabla 7.1.

	TAREA	HORAS
1	Planificación	35
2	Análisis	185
3	Diseño	195
4	Codificación	280
5	Pruebas	85
6	Despliegue	25
7	Seguimiento	12
TOTALES:		817

Tabla 7.1. Recursos Temporales por Fases del Proyecto

El diagrama de Gantt de la Figura 7.3 muestra la sucesión de cada una de las tareas que componen el desarrollo del proyecto.

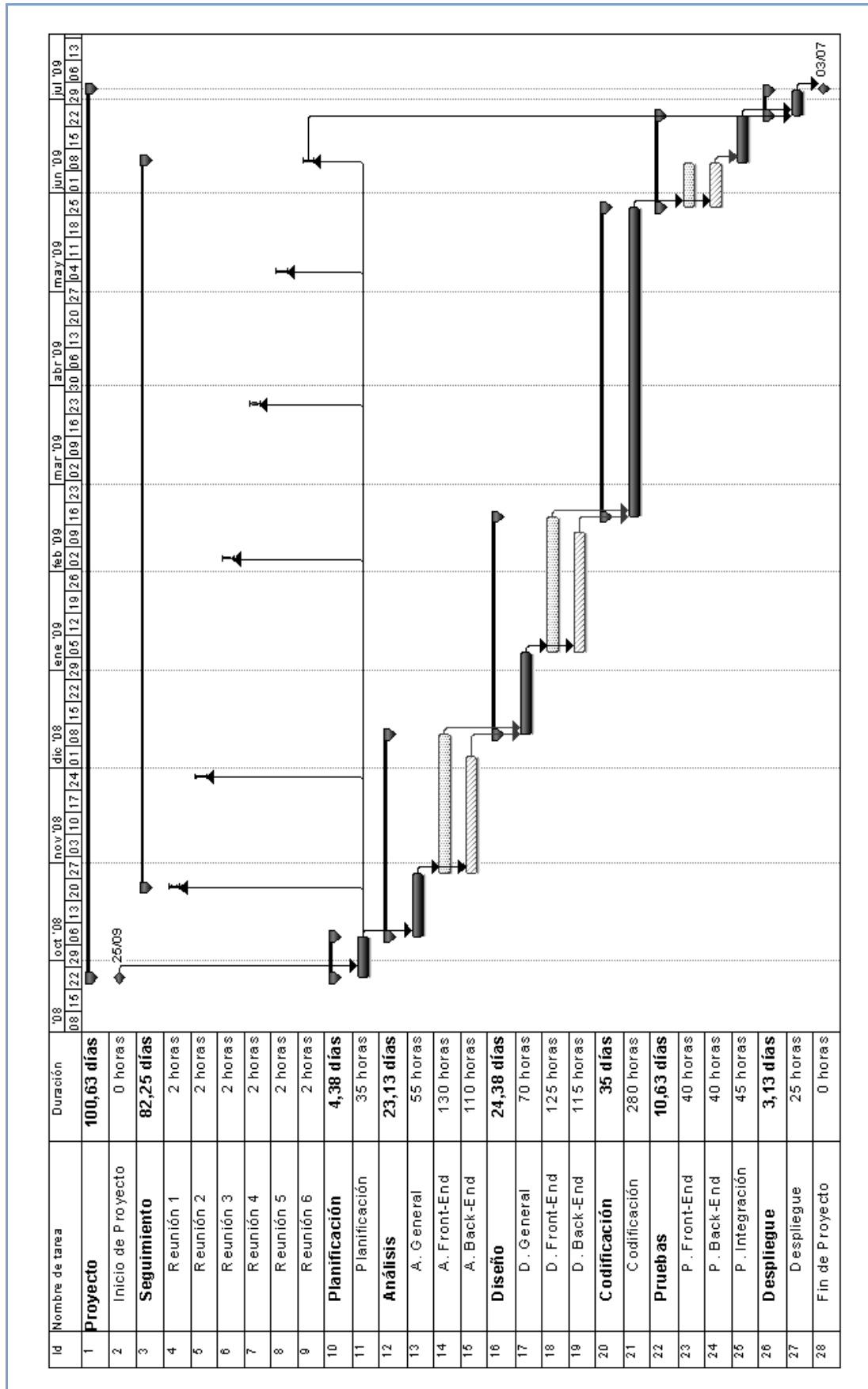


Figura 7.3. Diagrama de Gantt

7.4. ESTIMACIÓN DE RECURSOS ECONÓMICOS

En las siguientes secciones se determina el coste asociado a la ejecución del proyecto.

7.4.1. Recursos Materiales

La estimación del coste y de los periodos de amortización de los recursos materiales se han obtenido en base a lo descrito en [RD1777, 2004]. En este documento se establece un plazo de amortización de 8 años para los elementos de tratamiento de información y de 6 años para los programas informáticos. En base a esta información, el coste asociado a cada elemento viene determinado por la siguiente expresión.

$$\text{Coste} = \text{Precio} \cdot \text{Coef} = \text{Precio} \cdot \left(\text{Unidades} \cdot \frac{d}{a} \right)$$

Figura 7.4. Expresión para el Cálculo de las Amortizaciones

En la expresión anterior, a y d corresponden al periodo de amortización del elemento y a la duración del proyecto en años, respectivamente. La Tabla 7.2 muestra el desglose de los costes asociados a los recursos materiales.

CONCEPTO	PRECIO	COEFICIENTE	COSTE
PC Portátil Sony Vaio	974,64 €	9,4 %	91,37 €
Impresora HP D2460	41,16 €	9,4 %	3,86 €
Conexión ADSL a 6 Mbps durante 9 meses a 39,90 €/mes	359,10 €	100,0 %	359,10 €
Microsoft Office 2007	219,27 €	12,5 %	27,41 €
Microsoft Project 2007	149,00 €	12,5 %	18,63 €
Microsoft Visio 2007	221,00 €	12,5 %	27,63 €
Microsoft Visual Studio 2005	512,32 €	12,5 %	64,04 €
Macromedia Fireworks 2004	404,84€	12,5 %	50,61 €
TOTAL			642,64 €

Tabla 7.2. Recursos Materiales para el Desarrollo del Proyecto

7.4.2. Recursos Humanos

A partir de los recursos temporales presentados en la sección 7.3 y de los perfiles profesionales descritos a continuación, en la siguiente sección se

determina el coste asociado a los recursos humanos. En la Tabla 7.3 se muestran los perfiles de los profesionales requeridos para la ejecución de las diferentes tareas y los costes de cada uno de ellos.

DESCRIPCIÓN	€/HORA	HORAS	COSTE
Director de Proyecto	113,00 €	39	4.407,00 €
Jefe de Proyecto	75,00 €	77	5.775,00 €
Analista de Servicios	63,00 €	299	18.837,00 €
Desarrollador de Bases de Datos	50,00 €	112	5.600,00 €
Desarrollador de Aplicaciones	50,00 €	225	11.250,00 €
Diseñador Gráfico	50,00 €	25	1.250,00 €
Probador de Software	25,00 €	40	1.000,00 €
TOTAL			48.119,00 €

Tabla 7.3. Recursos Humanos

Los perfiles profesionales requeridos para desarrollar cada una de las tareas son los siguientes:

- **Director de Proyecto:** Profesional titulado en Ingeniería Informática con experiencia. Requiere habilidades como liderazgo, capacidad de comunicación e intermediación en conflictos. Los honorarios de este profesional se sitúan en 113 €/hora.
- **Jefe de Proyecto:** Profesional titulado en Ingeniería Informática con amplia formación en gestión de proyectos y recursos humanos. Los honorarios de este profesional se encuentran en 75 €/hora.
- **Analista de Servicios:** Profesional titulado en Ingeniería de Informática con conocimientos de arquitectura de redes, protocolos de comunicaciones y seguridad en entornos web. Así mismo, ha de poseer conocimientos en biometría. Los honorarios de este profesional se encuentra en 63 €/hora.
- **Desarrollador de Base de Datos:** Profesional titulado en Ingeniería Informática con conocimientos de dimensionado, parametrización y gestión de bases de datos. Requiere conocimientos en bases de datos SQL y otras tecnologías asociadas. Sus honorarios se encuentran en torno a los 50 €/hora.
- **Desarrollador de Aplicaciones:** Profesional titulado en Ingeniería Informática con conocimientos en ASP, ASP.NET, HTML, Java Script, CSS y otras tecnologías relacionadas con la elaboración de contenidos

Web. Sus honorarios se sitúan en 50 €/hora.

- Diseñador Gráfico: Profesional con título de Ciclo Formativo de Grado Superior en Artes Gráficas y amplios conocimientos de herramientas para diseño gráfico y maquetación como Freehand, Illustrator o Adobe Photoshop. El salario de este profesional se sitúa en 80 €/hora.
- Probador de Software: Profesional titulado en Ciclo Formativo de Grado Superior en Informática. Su función es la de validar la aplicación a nivel de usuario y reportar los problemas. Sus honorarios se encuentran en 25 €/hora.

7.4.3. Costes Totales

El coste total asociado al desarrollo del proyecto asciende a 84.397,41 € (ochenta y cuatro mil trescientos noventa y siete euros con cuarenta y un céntimos). De éstos, 81.272,53 € corresponden a la implementación del Front-End y 3.124,88 € a los costes soportados por el cliente. Las siguientes tablas recogen el desglose de los costes.

CONCEPTO	VALOR
Recursos Materiales	642,64 €
Recursos Humanos	48.119,00 €
Gastos Generales (20% Recursos Humanos)	9.623,80 €
Subtotal	58.385,44 €
Beneficios Empresariales (20% Subtotal)	11.677,09 €
Base Imponible	70.062,53 €
I.V.A. (16% Base Imponible)	11.210,00 €
TOTAL	81.272,53 €

Tabla 7.4. Costes del Proyecto

CONCEPTO	PRECIO/ UNIDAD	UNIDADES	COSTE
Servidor HP Proliant 155	2.863,00 €	1	2.863,00 €
Solicitud de Registro Dominio .es	26,88 €/año	1	26,88 €
Solicitud de Registro de Marca	235,00 €	1	235,00 €
TOTAL (I.V.A Incluido)			3.124,88 €

Tabla 7.5. Recursos Materiales Proporcionados por el Cliente

7.5. PLAN DEL PROYECTO

A lo largo del siguiente apartado se describe el plan del proyecto. En primer lugar se encuentra el plan de fases del proyecto, donde se describe cada una de las tareas que componen la planificación. El plan de proyecto continúa con el plan de modificaciones, que describe el mecanismo de control de las modificaciones requeridas por el desarrollo del proyecto. Para finalizar se incluye el plan de revisiones e informes, donde se analiza el modo de informar sobre el estado del proyecto y se definen las revisiones formales asociadas al avance del mismo.

7.5.1. Fases del Proyecto

El desarrollo del presente proyecto se ha dividido en un total de seis etapas. A continuación se particularizan los objetivos y las tareas a desarrollar en cada una de ellas.

Tarea 1. Planificación

La etapa de planificación pretende abordar los costes temporales y económicos de la ejecución del proyecto. Esta tarea se corresponde con la elaboración de la planificación contenida en el actual capítulo.

Tarea 2. Análisis

Durante la etapa de análisis se realiza una descripción completa de los requisitos funcionales del Front-End. La presente tarea se puede dividir en varias subtareas:

- Captura de los requisitos de usuario para el Front-End.
- Definición del álgebra del metamodelo con el fin de establecer el punto de partida en la interpretación y la semántica del lenguaje de modelado de los experimentos biométricos.
- Estudio de la arquitectura funcional de la plataforma y concretamente del Front-End.
- Definición de los perfiles de usuario presentes en la aplicación y los roles o funcionalidades asociadas a cada uno de ellos.
- Descripción de los casos de uso donde para proporcionar los escenarios de interacción del Front-End con los usuarios.

Tarea 3. Diseño

A lo largo de la tarea de diseño se efectúa la especificación de los componentes de la plataforma, la definición de los interfaces entre sistemas y la adopción de las tecnologías. Esta tarea se subdivide en varias fases.

- Definición del modelo de datos con el objetivo de establecer las entidades

presentes en la aplicación y las relaciones existentes entre ellas.

- Definición del interface de comunicación entre el Front-End y el Back-End.
- Establecimiento de la sintaxis soportada por el metamodelo.
- Definición del patrón arquitectónico adoptado para el Front-End y los patrones de diseño de los componentes que lo forman.
- Elaboración de los diagramas de navegación del Front-End para los diferentes perfiles de usuario.
- Diseño gráfico de los elementos visuales y estilos utilizados en el interface gráfico. Así mismo, se realiza la maquetación de los diferentes tipos de páginas existentes.
- Adopción de las tecnologías para implementar el Front-End.

Tarea 4. Codificación

La tarea de codificación aborda la implementación de las funcionalidades propuestas en etapas anteriores. La codificación se ha subdividido en varias etapas.

- Codificación de los sistemas necesarios para realizar operaciones de lectura, escritura y validación sobre los modelos de los experimentos biométricos.
- Codificación del sistema de comunicación entre el Front-End y el Back-End.
- Implementación de los elementos visuales y de navegación de la plataforma.
- Codificación del editor de experimentos.

Tarea 5. Pruebas

La fase de pruebas hace referencia a la evaluación del funcionamiento de la plataforma bajo entornos de ejecución genéricos. Dentro de la fase de pruebas se diferencian las siguientes etapas:

- Pruebas unitarias para verificar el correcto funcionamiento de cada uno de los módulos de código.
- Pruebas de integración para comprobar la interacción de todos los elementos unitarios que componen un proceso.
- Pruebas de integración entre el Front-End y el Back-End para verificar su correcto funcionamiento en el servidor.

- Pruebas de usabilidad y navegación para verificar que los elementos gráficos del Front-End responden de forma correcta a la interacción del usuario.

Tarea 6. Despliegue

La fase de despliegue se corresponde con la puesta en marcha de la plataforma en el servidor una vez que las pruebas se han realizado de forma satisfactoria.

7.5.2. Control de Modificaciones

Para realizar la codificación se ha empleado la herramienta Visual Studio de Microsoft. Este sistema ofrece la herramienta Source Safe [MSDN, 2009a], cuya función principal consiste en realizar un seguimiento de los historiales de los archivos y proyectos. La base de datos de Source Safe incluye todas las versiones de un archivo o proyecto y permite ver el historial para recuperar cualquier versión de estos elementos.

Visual Studio también posee un sistema de control de versiones llamado Team Foundation Server [MSDN, 2009b]. Éste ofrece colaboración en equipo, control de versiones, gestión de cambios y administración de la generación y elaboración de informes.

7.5.3. Seguimiento del Proyecto

A lo largo del proyecto se ha realizado un seguimiento quincenal de la evolución. Así mismo, se han propuesto reuniones mensuales para evaluar los objetivos alcanzados y el desarrollo conjunto del proyecto. En cada una de estas reuniones se ha planteado una revisión del trabajo realizado y los posibles cambios para mejorarlo.

7.6. HERRAMIENTAS

A continuación se presentan las herramientas utilizadas tanto para la implementación de la aplicación como para la gestión del proyecto y la redacción de la memoria.

Microsoft Visual Studio 2005




Microsoft Visual Studio ¹³ es un entorno de desarrollo integrado para sistemas basados en Windows. Soporta lenguajes de programación como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET y actualmente se están desarrollando extensiones para muchos otros lenguajes. Visual Studio permite crear aplicaciones, sitios y

¹³ <http://www.microsoft.com/spanish/msdn/vs2005/default.msp>

aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma. Visual Studio se ha utilizado para codificar las diferentes aplicaciones que forman la plataforma y para definir el metamodelo de experimentación. Para realizar la documentación del código generado se ha empleado la herramienta Sandcastle¹⁴, un plugin para Visual Studio que permite documentar código.

Microsoft Office 2007

 Microsoft Office 2007¹⁵ es la versión más reciente del paquete ofimático de Microsoft. Cuenta con herramientas para editar textos, realizar hojas de cálculo, presentaciones de diapositivas y otras muchas aplicaciones. La elaboración de la presente memoria se ha realizado mediante MS Word, para los diagramas UML se ha utilizado MS Visio, la planificación del proyecto se ha realizado con MS Project y los costes del proyecto se han calculado mediante MS Excel.

Macromedia Fireworks 2004



Fireworks¹⁶ es una aplicación en forma de estudio destinada al manejo híbrido de gráficos vectoriales y en mapa de bits. Permite la creación de prototipos de sitios web, interfaces de usuario e imágenes. Fireworks se ha utilizado principalmente en la creación y edición de los elementos visuales del Front-End.

Mozilla Firefox 3



Mozilla Firefox¹⁷ es un navegador de Internet libre y de código abierto desarrollado por la Corporación Mozilla, la Fundación Mozilla y un gran número de voluntarios externos. Firefox es un navegador multiplataforma y está disponible en versiones para Microsoft Windows, Mac OS X, GNU/Linux y algunos sistemas basados en Unix. Firefox se ha empleado para evaluar el funcionamiento de la aplicación web de la plataforma.

Mendeley



Mendeley¹⁸ es una aplicación que permite organizar referencias bibliográficas y generar citas en los formatos especificados de una manera elegante y estándar. Ofrece integración con MS Office Word para insertar citas desde una librería personal o directamente desde cualquier documento abierto. Así mismo, permite acceder a las referencias de la librería

¹⁴ <http://www.codeplex.com/Sandcastle>

¹⁵ <http://office.microsoft.com/es-es/default.aspx>

¹⁶ <http://www.adobe.com/products/fireworks/>

¹⁷ <http://www.mozilla-europe.org/es/firefox/>

¹⁸ <http://www.mendeley.com/>

desde el sitio web para compartirlas con otros usuarios. Mendeley se ha utilizado en la gestión de las referencias bibliográficas del presente documento.

CONCLUSIONES Y TRABAJO FUTURO

Este último capítulo presenta los resultados obtenidos y los compara con los objetivos planteados inicialmente. Así mismo, ofrece las líneas futuras de trabajo sobre la plataforma desarrollada.

8.1. CONCLUSIONES

Las conclusiones presentadas a continuación resultan de la confrontación entre los objetivos planteados al inicio del proyecto y los conseguidos tras la finalización del mismo.

Una de las tareas más difíciles a la hora de desarrollar un sistema software es establecer las condiciones de satisfacción para que el trabajo se dé por concluido. Para establecer el grado de éxito del proyecto se recuerdan los objetivos generales de la plataforma y se muestra el grado de dificultad de la consecución de cada uno de ellos. Los objetivos cubiertos con la implementación del Front-End son los siguientes:

- Se ha desarrollado un metamodelo capaz de describir las operaciones y los datos de los experimentos biométricos.
- Se ha desarrollado una aplicación que permite crear y modificar experimentos biométricos de acuerdo al metamodelo propuesto.
- Se ha implementado un sistema capaz de generar descripciones de datos biométricos para bases de datos MySQL, SQL Server y Access.
- Se ha realizado un sistema capaz de generar descripciones de algoritmos basados en librerías de código para el entorno de ejecución .NET.
- Se ha logrado desarrollar una aplicación que salvaguarda la información sensible de las muestras biométricas empleadas en la ejecución de los experimentos.
- Se ha construido un sistema capaz de presentar informes de prestaciones a partir de los resultados obtenidos tras la ejecución de los experimentos.
- Por último, se ha desarrollado una arquitectura de comunicación para

permitir la comunicación entre los sistemas de la plataforma asegurando la independencia de los mismos.

Un punto clave en el desarrollo del proyecto ha sido el trabajo en equipo. Gran parte de los subsistemas de la plataforma han requerido un fuerte trabajo en equipo para conseguir su implementación. El trabajo en equipo ha sido clave en las etapas de análisis y diseño, especialmente en el desarrollo del interface de comunicación y en la definición del metamodelo.

Otro punto clave en el desarrollo del proyecto ha sido la reutilización de código. Tanto el Front-End como el Back-End comparten un número importante de operaciones. Por este motivo, se han desarrollado conjuntamente una serie de librerías comunes con todas aquellas operaciones compartidas por los dos subsistemas de la plataforma.

El grado de cumplimiento de los objetivos planteados al inicio puede considerarse satisfactorio. La integración de las diferentes aplicaciones que componen la plataforma va a permitir la realización de experimentos bajo un entorno de experimentación común.

8.2. TRABAJOS FUTUROS

En esta sección se enumeran algunas propuestas para trabajos futuros sobre el Front-End:

- Agregar nuevas operaciones primitivas al metamodelo con el fin de ampliar las operaciones sobre los datos biométricos.
- Permitir la descripción de datos biométricos a partir de nuevos tipos de bases de datos.
- Mejorar los mecanismos de seguridad en la comunicación entre la aplicación web y el editor de experimentos.
- Ofrecer un mecanismo de intercambio de resultados entre las entidades de investigación con el objetivo de ofrecer un marco de comparación de resultados.

REFERENCIAS

[Amber, 2004]

S. Amber: "The Object Primer. Agile Model Driven Development with UML 2". 3rd Ed. Cambridge University Press. 2004.

[Atkinson et al, 2003]

C. Atkinson and T. Kühne: "Model-Driven Development. A meta-modeling foundation". IEEE Software, May 2003, pp. 36-41. 2003.

[Atzeni et al, 1998]

P. Atzeni, G. Mecca and P. Merialdo: "Design and maintenance of data-intensive web sites". Advances in Database Technology, EDBT '98. LNCS, vol. 1377, pp. 436-450. Springer. 1998.

[Blackburn, 2004]

D. Blackburn: "Biometrics 101". Federal Bureau of Investigation. 2004.

[Boehm et al, 2007]

B.W. Boehm and R.W. Selby: "Software engineering: Barry W. Boehm's lifetime contributions to software development, management, and research". Ed. Wiley-IEEE. 2007.

[Bolle et al, 2003]

R. M. Bolle, J. H. Connell, S. Pankanti, N. K. Ratha, and A. W. Senior: "Guide to Biometrics". New-York: Springer-Verlag. 2003.

[Boyd et al, 2005]

J.E. Boyd and J.J. Little: "Biometric Gait Recognition". LNCS, vol. 3161, pp.19-42. Springer. 2005.

[Buschmann, 1996]

F. Buschmann: "Pattern-Oriented Software Architecture. A System of Patterns". Ed. John Wiley. 1996.

[Capelli et al, 2006]

R. Cappelli, D. Maio, D. Maltoni, J.L. Wayman and A.K. Jain.: "Performance evaluation of fingerprint verification systems". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no.1, pp. 3-18. 2006.

[Chang et al, 1993]

H.D. Chang, J.F. Wang, and H.M. Suen: "Dynamic handwritten Chinese signature verification". Proceedings of Second IEEE International Conference on Document Analysis and Recognition, pp. 258-261. 1993.

[Codd, 1970]

E. Codd: "A relational model of data for large shared data banks". Communications of the ACM, vol. 13, no. 6, pp. 377-387. IBM Research Laboratory, San Jose, California. 1970.

[Cuevas, 2003]

A. Cuevas: "Gestión del proceso software". Centro de Estudios Ramón Areces, S.A. 2003

[Dessimoz et al, 2005]

D. Dessimoz, J. Richiardi, C. Champod and A. Drygajlo: "Multimodal biometrics for identity documents". University of Lausanne and EPFL (European Biometrics Portal). Technical Report PFS 341-08.05. 2005.

[Duta et al, 2002]

N. Duta, A.K. Jain and K.V. Mardia: "Matching of palm print". Pattern Recognition Letters, vol. 23, pp. 477-485. 2002.

[ESA, 1991]

ESA: "Software Engineering Standards. Issue 2". ESA Board for Software Standardization and Control. European Space Agency. 1991.

[Fernández, 2001]

A. Fernández: "Introducción a UML". Universidad de Vigo. 2001.

Disponible en: <http://www-gris.det.uvigo.es/~avilas/UML/UML.html>

[Fogarty et al, 2005]

J. Fogarty, R. Baker and S. Hudson: "Case studies in the use of ROC curve analysis for sensor-based estimates in human computer interaction". Proceedings of the ACM International Conference Series. 2005.

[Gamma, 1998]

E. Gamma: "Design Patterns. Elements of Reusable Object-Oriented Software". Ed. Addison Wesley. 1998.

[Jain et al, 2004]

A.K. Jain, A. Ross and S. Prabhakar: "An introduction to biometric recognition". IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image and Video Based Biometrics, vol. 14, no. 1, pp. 4-20. 2004.

[Jain et al, 2005]

A.K. Jain, K. Nandakumar and A. Ross: "Score normalization in multimodal biometric systems". Pattern Recognition, vol. 38, no. 12, pp. 2270-2285. 2005.

[Justino et al, 2001]

E. Justino, F. Bortolozzi and R. Sabourin: "Off-line signature verification using HMM for random, simple and skilled forgeries". Proceedings of 6th International Conference on Document Analysis and Recognition, pp. 1031-1034. 2001.

[Hong et al, 1999]

L. Hong, A.K. Jain and S. Pankanti: "Can multibiométricos improve performance?" Proceedings of AutoID '99, pp. 59-64. 1999.

[Huang et al, 1997]

Kai Huang and Hong Yan: "Off-line signature verification based on geometric feature extraction and neural network classification". Pattern Recognition, vol. 30, no. 1, pp. 9-17. 1997.

[Humphrey, 1995]

W.S. Humphrey: "A discipline for software engineering". Ed. Addison Wesley. 1995.

[IEEE, 1998]

IEEE Standard 830-1998: "IEEE Recommended Practice for Software Requirements Specifications".

Disponible en: <http://standards.ieee.org/reading/ieee/std/se/830-1998.pdf>

[Kobrun, 2001]

C. Kobrun: "Introduction to UML: Structural and use case modeling". Object Modeling with OMG UML Tutorial Series. 2001

[Larman, 2003]

C. Larman: "UML y patrones". Ed. Pearson Educación. 2ª ed. 2003.

[LOPD, 1999]

Jefatura del Estado: "Ley Orgánica 15/1999, de 13 de Diciembre, de Protección de Datos de Carácter Personal". Boletín Oficial del Estado. 1999

Disponible en: http://www.boe.es/g/es/bases_datos/doc.php.

[Maltoni et al, 2003]

D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar: "Handbook of Finger-print Recognition". Springer-Verlag. 2003.

[Mansfield et al, 2002]

A.J. Mansfield and J.L. Wayman: "Best practices in testing and reporting performances of biometric devices". Centre for Mathematics and Scientific Computing. National Physical Laboratory. 2002.

[Martin et al, 2004]

A. Martin, M. Przybocki, and J. Campbell: "The NIST speaker recognition evaluation program". Biometric Systems Technology, Design and Performance Evaluation, J. Wayman, A. Jain, D. Maltoni, and D. Maio, Eds. Springer-Verlag. 2004.

[Masek, 2003]

L. Masek: "Recognition of human iris patterns for biometric identification". Bachelor of Engineering degree, School of Computer Science and Software Engineering, University of Western Australia, 2003.

[Monrose et al, 2000]

F. Monrose A.D. Rubin: "Keystroke dynamics as a biometric for authentication". Future Generation Computer Systems, vol. 16, pp. 351-359. Elsevier. 2000.

[Morville, 2006]

P. Morville: "Information architecture for the World Wide Web. Designing Large-Scale Web Sites". O'Reilly Media. 2006.

[MSDN, 2007]

MSDN: "Fundamentos de la normalización de bases de datos". Microsoft Ayuda y Soporte. 2007.

Disponible en: <http://support.microsoft.com/kb/283878>

[MSDN, 2009a]

MSDN: "Documentación de Visual Studio 2005. Control de Versiones". 2009.

Disponible en: [http://msdn.microsoft.com/es-es/library/shyhfx67\(vs.80\).aspx](http://msdn.microsoft.com/es-es/library/shyhfx67(vs.80).aspx)

[MSDN, 2009b]

MSDN: "Documentación de Visual Studio Team System". 2009.

Disponible en: [http://msdn.microsoft.com/es-es/library/fda2bad5\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/fda2bad5(VS.80).aspx)

[MSDN, 2009c]

MSDN: "Interprocess Communications. Pipes". 2009.

Disponible en: [http://msdn.microsoft.com/en-us/library/aa365780\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa365780(VS.85).aspx)

[MSDN, 2009d]

MSDN: "Interprocess Communications. Named Pipes". 2009.

Disponible en: [http://msdn.microsoft.com/en-us/library/aa365590\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa365590(VS.85).aspx)

[Nielsen, 2000]

J. Nielsen: "Designing web usability". New Riders. 2000.

[Nixon et al, 2006]

M.S. Nixon, T.N. Tan and R. Chellappa: "Human identification based on gait". International Series on Biometrics, Springer. 2006.

[Oglesby et al, 1988]

J. Oglesby and J. Mason: "Speaker recognition with a neural classifier". Speech 88: Proceedings of the 7th Federation of Acoustical Societies of Europe, pp. 1357-1363. 1988.

[OMG, 2009]

OMG: "Unified Modeling Language specifications". 2009.

Disponible en: <http://www.omg.org/spec/UML/2.2/>

[Pandit et al, 1998]

M. Pandit and J. Kittler: "Feature selection for a DTW-based speaker verification system". IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1998, vol. 2, pp. 769-772. 1998.

[Phillips et al, 2000]

P.J Phillips, A. Martin, C.L. Wilson and M. Przybocky: "An introduction to evaluating biometric systems". Computer, vol. 33, no. 2. 2000.

[Phillips et al, 2003]

P.J. Phillips, P. Grother, R.J. Michels, D.M. Blackburn, E. Tabassi and J.M. Bone: "Facial recognition vendor test 2002, evaluation report". 2003.

Disponible en: <http://www.frvt.org/FRVT2002>

[Piattini et al, 2006]

M. Piattini, E. Marcos, B. Calero B. Vela: "Tecnología y diseño de bases de datos". RA-MA. 2006.

[Poritz, 1982]

A. Poritz: "Linear predictive Hidden Markov Models and the speech signal". IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1982, vol. 7, pp. 1291-1294. 1982.

[Prabhakar et al, 2003]

S. Prabhakar, S. Pankanti, and A. K. Jain: "Biometric recognition: Security and privacy concerns". IEEE Security and Privacy, vol. 1, no. 2, pp. 33-42. 2003.

[Puente, 2007]

L. Puente: "BEEP-MM. Propuesta de un metamodelo de experimentación biométrica".

[Puente et al, 2008]

L. Puente Rodríguez, A. García Crespo, M. J. Poza Lara and B. Ruiz Mezcuca: "Study of different fusion techniques for multimodal biometric authentication". Proceedings of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication. pp. 666-671. 2008.

[Raphael et al, 1974]

D. E. Raphael and J.R. Young: "Automated Personal Identification". SRI International. 1974.

[RD1777, 2004]

Ministerio de Economía y Hacienda: "Real Decreto 1777/2004, de 30 de Julio, por el que se aprueba el Reglamento del impuesto de Sociedades. Anexo: Tablas de Coeficientes de Amortización". 2004.

Disponible en: http://www.boe.es/g/es/bases_datos/doc.php

[Renesse, 2002]

R. L. van Renesse: "Implications of applying biometrics to travel documents". Proceedings of SPIE, vol. 4677, pp. 290-298. 2002.

[Reynolds, 1995]

D. A. Reynolds: "Speaker identification and verification using gaussian mixture speaker models". *Speech Communication*, vol. 17, pp. 91-108. 1995.

[Richiardi et al, 2003]

J. Richiardi and A. Drygajlo: "Gaussian mixture models for on-line signature verification". *Proceedings of International Multimedia Conference 2003*, pp. 115-122. 2003.

[Ruíz, 1998]

B. Ruíz: "Modelado estadístico y conexionista para reconocimiento de locutores con aprendizaje de la variabilidad temporal del habla". *Tesis Doctoral*. 1998.

[Saake et al, 1993]

G. Saake, R. Jungclaus and T. Hartmann: "Application modeling in heterogeneous environments using an object specification language". *Proceedings of CoopIS*, pp. 309-318. 1993.

[Sato et al, 1982]

Y. Sato and K. Kogure: "Online signature verification based on shape, motion, and writing pressure". *Proceedings of 6th International Conference on Pattern Recognition*, pp. 823-826. 1982.

[Sánchez et al, 2000]

R. Sánchez Reillo, C. Sánchez Avila and A. González Marcos: "Biometric identification through hand geometry measurements". *IEEE Transactions on Pattern Analysis and Matching*, vol. 22, no. 10, pp. 1168-1171. 2000.

[Sánchez, 2000]

R. Sánchez Reillo: "Hand geometry pattern recognition through Gaussian Mixture Modeling". *Proceedings of the 15th Int. Conf. Pattern Recognition*, vol. 2, pp. 941-944. 2000.

[Schmidt et al, 1996]

M. Schmidt and H. Gish: "Speaker identification via support vector classifiers". *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1996*, vol. 1, pp. 105-108. 1996.

[Seidewitz, 2003]

E. Seidewitz: "What models mean". *IEEE Software*, September 2003, pp. 26-32. 2003.

[Silberschatz et al, 2006]

A. Silberschatz, P.B. Galvin and G. Gagne: "Fundamentos de sistemas operativos". Mc Graw Hill. 2006.

[Sirovich et al, 1987]

L. Sirovich and M. Kirby: "Low-dimensional procedure for the characterization of human faces". Journal of the Optical Society of America, vol. 4, no. 3, pp. 519-524. 1987.

[Tisee, 2003]

C.L. Tisse : "Contribution à la vérification biométrique de personnes par reconnaissance de l'iris". Ph. D, Université de Montpellier. 2003.

[Van Duyne et al, 2002]

D.K. Van Duyne, J.A. Landay and J.I. Hong: "The design of sites. Patterns, principles and processes for crafting a customer-centered web experience". Addison Wesley. 2002.

[W3C, 2004]

W3C Recommendation: "XML Information Set". 2nd Ed. 2004.

Disponible en: <http://www.w3.org/TR/xml-infoset/>

[Wildes, 2005]

R. Wildes: "Iris recognition". Biometric Systems: Technology, Design and Performance Evaluation, J. L. Wayman, A. K. Jain, D. Maltoni, and D. Maio, Eds. London: Springer-Verlag, pp. 63-95. 2005.

[Wolff, 1993]

K. E. Wolff: "A first course in formal concept analysis". Fachhochschule Darmstadt. Forschungsgruppe Begriffsanalyse der Technischen Hochschule Darmstadt. 1993.

[Yager et al, 2004]

N. Yager and A. Amin: "Fingerprint verification based on minutiae features: a review". Pattern Analysis and Application, vol. 17, pp. 94-113. 2004.

[Zhao et al, 2003]

W. Zhao, R. Chellappa, J. Phillips, and A. Rosenfeld: "Face recognition: A literature survey". ACM Computing Surveys, vol. 35, no. 4, pp. 399-458. 2003.

ANEXO A. GLOSARIO

A lo largo de las siguientes páginas se presentan las definiciones de los conceptos utilizados a lo largo de la memoria.

Almacén de Información Biométrica

Tabla desde la que se extraen o insertan datos durante la ejecución de un experimento.

Caducidad

Concepto que hace referencia al nivel de actividad de las entidades de investigación. De forma general, una entidad está no caducada puesto que se produce el acceso a la misma por parte de sus miembros de forma continua y en plazos cortos de tiempo. Una entidad pasa a estar caducada cuando no se produce ninguna actividad durante un determinado periodo de tiempo y ningún usuario accede a ella. La caducidad de las entidades se establece para controlar los recursos de la plataforma y mantener el servidor libre de datos inservibles, de tal forma que aquellas entidades que no demuestran actividad alguna son eliminadas junto a sus espacios de trabajo.

Conexión

Define la cadena de conexión a la base de datos que contiene la información de un almacén.

Definición de Experimento

Archivo XML que contiene el modelo de un experimento biométrico.

Descriptor de Contenido

Archivo XML que contiene la descripción de los campos y tipos de datos asociados a un almacén de información biométrica.

Descriptor de Interfaces

Archivo XML que define los parámetros de entrada, salida y configuración de los procesos de tratamiento de información biométrica.

Entidad de Investigación

Organización con fines de investigación externa a la plataforma que utiliza los recursos proporcionados por ésta. Una entidad investigadora está formada por grupos de trabajo y por usuarios.

Espacio de Trabajo

El espacio de trabajo hace referencia a los recursos de almacenamiento que la plataforma ofrece a cada una de las entidades. En el espacio de trabajo, una entidad investigadora puede almacenar procesos, experimentos, datos biométricos y los informes de resultados de la ejecución de los experimentos. El espacio de trabajo se comparte por todos los grupos dependientes de la entidad.

Grupo de Trabajo

Agrupación de usuarios de una entidad de investigación que posee acceso común a una serie de experimentos e informes de resultados.

Librería

Archivo de tipo DLL que contiene clases para su utilización en procesos.

Proceso

Conjunto de clases, obtenidas de una o más librerías de código, que transforman los datos biométricos que reciben.

ANEXO B. ESPECIFICACIÓN DE REQUISITOS

El presente anexo recoge la especificación de requisitos funcionales del sistema. En primer lugar, se ofrecen los requisitos generales del sistema y a continuación se presentan los requisitos de usuario.

ID	ESENCIAL	MODIFICABLE	PRIORIDAD
RG-001	Si	No	Alta
DESCRIPCIÓN: El sistema incorpora un subsistema de autenticación de usuarios con el fin de restringir el acceso a la aplicación.			
RG-002	Si	No	Alta
DESCRIPCIÓN: La aplicación es multiusuario.			
RG-003	Si	No	Alta
DESCRIPCIÓN: La aplicación envía notificaciones de ejecución al Back-End de la plataforma.			
RG-004	Si	No	Alta
DESCRIPCIÓN: La aplicación genera e interpreta modelos de experimentos conforme a la especificación del metamodelo.			

ID	ESENCIAL	MODIFICABLE	PRIORIDAD
RG-005	Si	No	Alta
DESCRIPCIÓN: El sistema ofrece un mecanismo de gestión de experimentos para controlar el acceso concurrente de los usuarios.			
RG-006	Si	No	Alta
DESCRIPCIÓN: El sistema ofrece un mecanismo de gestión de la cola de ejecución con el fin de controlar el acceso a los recursos del Back-End.			
RU-001	Si	No	Alta
DESCRIPCIÓN: La aplicación permite valorar las propuestas de creación de entidades de investigación antes de la creación de las mismas.			
RU-002	Si	No	Alta
DESCRIPCIÓN: El sistema permite crear y eliminar entidades de investigación. Así mismo, permite la modificación de su información.			
RU-003	No	No	Media
DESCRIPCIÓN: La aplicación notifica a los usuarios propietarios de la expiración de sus entidades.			
RU-004	Si	No	Alta
DESCRIPCIÓN: La aplicación permite eliminar las entidades de investigación sin actividad.			
RU-005	Si	No	Alta
DESCRIPCIÓN: El sistema ofrece un espacio de trabajo privado a cada entidad de investigación			

ID	ESENCIAL	MODIFICABLE	PRIORIDAD
RU-006	Si	No	Alta
DESCRIPCIÓN: Cada entidad de investigación tiene al menos un usuario propietario.			
RU-007	Si	No	Alta
DESCRIPCIÓN: La aplicación permite crear y eliminar cuentas de usuario. De la misma forma, permite al usuario modificar su información.			
RU-008	No	No	Baja
DESCRIPCIÓN: Las entidades de investigación poseen un número máximo de miembros.			
RU-009	Si	No	Alta
DESCRIPCIÓN: La aplicación permite al usuario recordar la contraseña de acceso al sistema.			
RU-010	No	Si	Baja
DESCRIPCIÓN: La aplicación permite crear y eliminar mensajes de comunicación en el entorno de la aplicación (públicos).			
RU-011	No	Si	Baja
DESCRIPCIÓN: La aplicación permite crear y eliminar mensajes de comunicación en el entorno de la entidad de investigación (privados).			
RU-012	No	Si	Baja
DESCRIPCIÓN: La aplicación permite recoger las opiniones y los comentarios de los usuarios con el fin de obtener feed-back.			

ID	ESENCIAL	MODIFICABLE	PRIORIDAD
RU-013	Si	No	Alta
DESCRIPCIÓN: La aplicación permite la creación y la eliminación de grupos de trabajo dentro de las entidades de investigación.			
RU-014	No	No	Baja
DESCRIPCIÓN: Las entidades de investigación poseen un número máximo de grupos de trabajo.			
RU-015	Si	No	Alta
DESCRIPCIÓN: La aplicación permite agregar y eliminar miembros de la entidad de investigación a los diferentes grupos de trabajo. Así mismo, permite establecer los permisos de los miembros dentro de cada grupo.			
RU-016	Si	No	Alta
DESCRIPCIÓN: El sistema ofrece la posibilidad de modificar los privilegios de los miembros de la entidad de investigación.			
RU-017	Si	No	Alta
DESCRIPCIÓN: La aplicación permite definir modelos de experimentos biométricos de acuerdo al metamodelo establecido.			
RU-018	No	No	Media
DESCRIPCIÓN: El sistema permite generar diferentes versiones de la misma definición de un experimento.			
RU-019	Si	No	Alta
DESCRIPCIÓN: El sistema permite eliminar definiciones de experimentos.			

ID	ESENCIAL	MODIFICABLE	PRIORIDAD
RU-020	Si	No	Alta
DESCRIPCIÓN: La definición de un experimento solamente puede pertenecer a un grupo de trabajo.			
RU-021	Si	No	Alta
DESCRIPCIÓN: El sistema permite poner en la cola de ejecución las definiciones de los experimentos biométricos.			
RU-022	No	No	Media
DESCRIPCIÓN: Las entidades de investigación tienen un número máximo de experimentos que pueden ser puestos de forma simultánea en la cola de ejecución.			
RU-023	No	No	Media
DESCRIPCIÓN: La aplicación notifica al usuario del comienzo y del final de la ejecución de los experimentos.			
RU-024	Si	No	Alta
DESCRIPCIÓN: El sistema permite incorporar y eliminar librerías de código para los procesos. Así mismo, permite la modificación de su información.			
RU-025	Si	No	Alta
DESCRIPCIÓN: La aplicación acepta librerías de código de tipo DLL, tanto de código gestionado como no gestionado.			
RU-026	Si	No	Alta
DESCRIPCIÓN: La aplicación permite crear procesos para el tratamiento de los datos biométricos a partir de las librerías de código disponibles.			

ID	ESENCIAL	MODIFICABLE	PRIORIDAD
RU-027	Si	No	Media
DESCRIPCIÓN: El sistema ofrece procesos públicos a las entidades de investigación.			
RU-028	Si	No	Alta
DESCRIPCIÓN: La aplicación permite crear y eliminar almacenes de información biométrica. Así mismo, permite modificar su información.			
RU-029	Si	Si	Alta
DESCRIPCIÓN: El sistema es capaz de generar descripciones de almacenes de información a partir de tablas MySQL, SQL Server y Access.			
RU-030	Si	No	Media
DESCRIPCIÓN: El sistema es capaz de generar las descripciones de los almacenes generados tras la ejecución de un experimento.			
RU-031	Si	No	Alta
DESCRIPCIÓN: La aplicación permite crear y eliminar conexiones a bases de datos para obtener nuevos almacenes de información.			
RU-032	Si	No	Alta
DESCRIPCIÓN: El sistema ofrece almacenes públicos de información a las entidades de investigación.			
RU-033	Si	No	Alta
DESCRIPCIÓN: Un almacén de información solamente puede encontrarse en la localización establecida por su conexión.			

ID	ESENCIAL	MODIFICABLE	PRIORIDAD
RU-034	No	No	Media
DESCRIPCIÓN: La aplicación permite la utilización de almacenes de información privados en los modelos de los experimentos.			
RU-035	Si	No	Alta
DESCRIPCIÓN: El sistema es capaz de generar informes de prestaciones tras la ejecución de un experimento biométrico.			
RU-035	Si	No	Media
DESCRIPCIÓN: La ejecución de un experimento genera como mínimo un informe de resultados.			
RU-036	SI	No	Media
DESCRIPCIÓN: La aplicación permite descargar los informes de resultados.			
RU-037	Si	No	Media
DESCRIPCIÓN: La eliminación de un experimento conlleva la eliminación de todos sus informes; sin embargo, la eliminación de un informe no conlleva la eliminación del experimento.			
RU-037	No	No	Media
DESCRIPCIÓN: La aplicación permite consultar y modificar los parámetros de configuración de la plataforma.			

ANEXO C. SEGUIMIENTO DEL PROYECTO

ACTA DE REUNIÓN 1 (24/10/2008)

ASISTENTES:

- Director de Proyecto: Luis Puente Rodríguez
- Jefe del proyecto Back-End: Elena Arribas González
- Jefe del proyecto Front-End: Diego Carrero Figueroa

ORDEN DEL DÍA:

1. Captura de los requisitos funcionales de la plataforma.
2. Definición de los requisitos del metamodelo.
3. Definición del alcance del Front-End.

DESARROLLO DE LA SESIÓN:

1. Se definieron los objetivos de la plataforma y se establecieron las restricciones impuestas a su implementación.
2. Se recogió la propuesta del metamodelo específico para el dominio de experimentación.
3. Se plantearon los objetivos a conseguir con la elaboración del proyecto.

ACTA DE REUNIÓN 2 (28/11/2008)**ASISTENTES:**

- Director de Proyecto: Luis Puente Rodríguez
- Jefe del proyecto Back-End: Elena Arribas González
- Jefe del proyecto Front-End: Diego Carrero Figueroa

ORDEN DEL DÍA:

1. Entrega del documento del alcance de la aplicación.
2. Revisión de los requisitos funcionales de la plataforma y del metamodelo.
3. Elección del ciclo de vida del proyecto.
4. Análisis del modelo de datos del sistema.
5. Definición del interface de comunicación entre los subsistemas.

DESARROLLO DE LA SESIÓN:

1. Se hizo entrega del documento de Alcance del Front-End quedando su aprobación pendiente para la siguiente reunión.
2. Se aprobaron los requisitos funcionales propuestos para los subsistemas de la plataforma.
3. Se optó por el ciclo de vida iterativo incremental debido a la complejidad del proyecto.
4. Se presentó el análisis conceptual de alto nivel sobre la base de datos de la plataforma y se eligió la utilización de bases de datos independientes para el sistema y las entidades.
5. Se definió el interface de comunicación con el objetivo de asegurar la independencia de los subsistemas de la plataforma.

ACTA DE REUNIÓN 3 (05/02/2009)

ASISTENTES:

- Director de Proyecto: Luis Puente Rodríguez
- Jefe del proyecto Back-End: Elena Arribas González
- Jefe del proyecto Front-End: Diego Carrero Figueroa

ORDEN DEL DÍA:

1. Revisión del alcance del proyecto.
2. Entrega del documento de análisis.
3. Presentación de los documentos relativos a la gestión y planificación del proyecto.

DESARROLLO DE LA SESIÓN:

1. Se aprobó el alcance del proyecto del Front-End.
2. Se entregó el documento de Análisis del Front-End.
3. Se presentaron los documentos de gestión del proyecto.

ACTA DE REUNIÓN 4 (26/03/2009)**ASISTENTES:**

- Director de Proyecto: Luis Puente Rodríguez
- Jefe del proyecto Back-End: Elena Arribas González
- Jefe del proyecto Front-End: Diego Carrero Figueroa

ORDEN DEL DÍA:

1. Revisión del análisis funcional.
2. Presentación del diseño.
3. Definición del contenido de la memoria del proyecto.
4. Resolución de dudas.

DESARROLLO DE LA SESIÓN:

1. Se aprobó el análisis funcional del Front-End.
2. Se establecieron las pautas de diseño para el Front-End, se hizo entrega del documento de la definición del Metamodelo y se eligieron las tecnologías para la implementación.
3. Se realizó un esbozo del contenido de la memoria y cómo debía ser tratado cada uno de sus apartados.
4. Se evaluó el estado del proyecto y debido a retrasos en la planificación se decidió reajustar la misma.

ACTA DE REUNIÓN 5 (07/05/2009)**ASISTENTES:**

- Director de Proyecto: Luis Puente Rodríguez
- Jefe del proyecto Back-End: Elena Arribas González
- Jefe del proyecto Front-End: Diego Carrero Figueroa

ORDEN DEL DÍA:

1. Entrega del documento de diseño.
2. Definición de aspectos referentes a la implementación.

DESARROLLO DE LA SESIÓN:

1. Se hizo entrega del documento de Diseño del Back-End y su evaluación quedó pendiente para la siguiente reunión.
2. Se eligió la implementación del interface de comunicación mediante una tubería con nombre y se decidió el uso de librerías comunes entre los subsistemas con el fin de reutilizar código.

ACTA DE REUNIÓN 6 (11/06/2009)**ASISTENTES:**

- Director de Proyecto: Luis Puente Rodríguez
- Jefe del proyecto Back-End: Elena Arribas González
- Jefe del proyecto Front-End: Diego Carrero Figueroa

ORDEN DEL DÍA:

1. Revisión del documento de Diseño del Front-End.
2. Presentación de resultados sobre la integración de los dos subsistemas.

DESARROLLO DE LA SESIÓN:

1. Se aprobó el diseño del Front-End.
2. Se propuso solución a los distintos problemas que surgieron durante la integración de los subsistemas.

